THE COPY CORNER DATABASE

Lowell Olson
Micah Gemmell
Mark Griffith

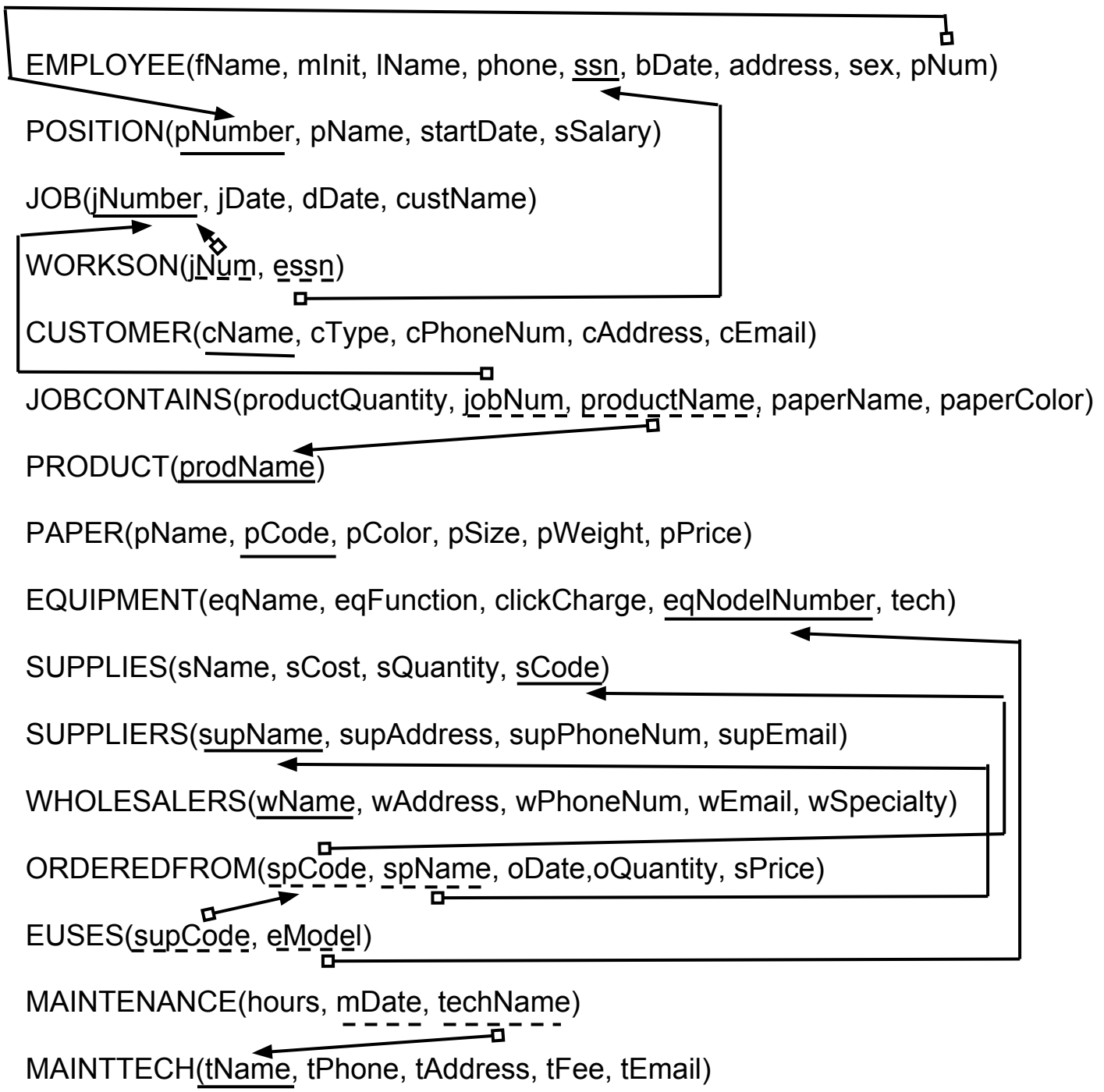
CS355
3/13/2013

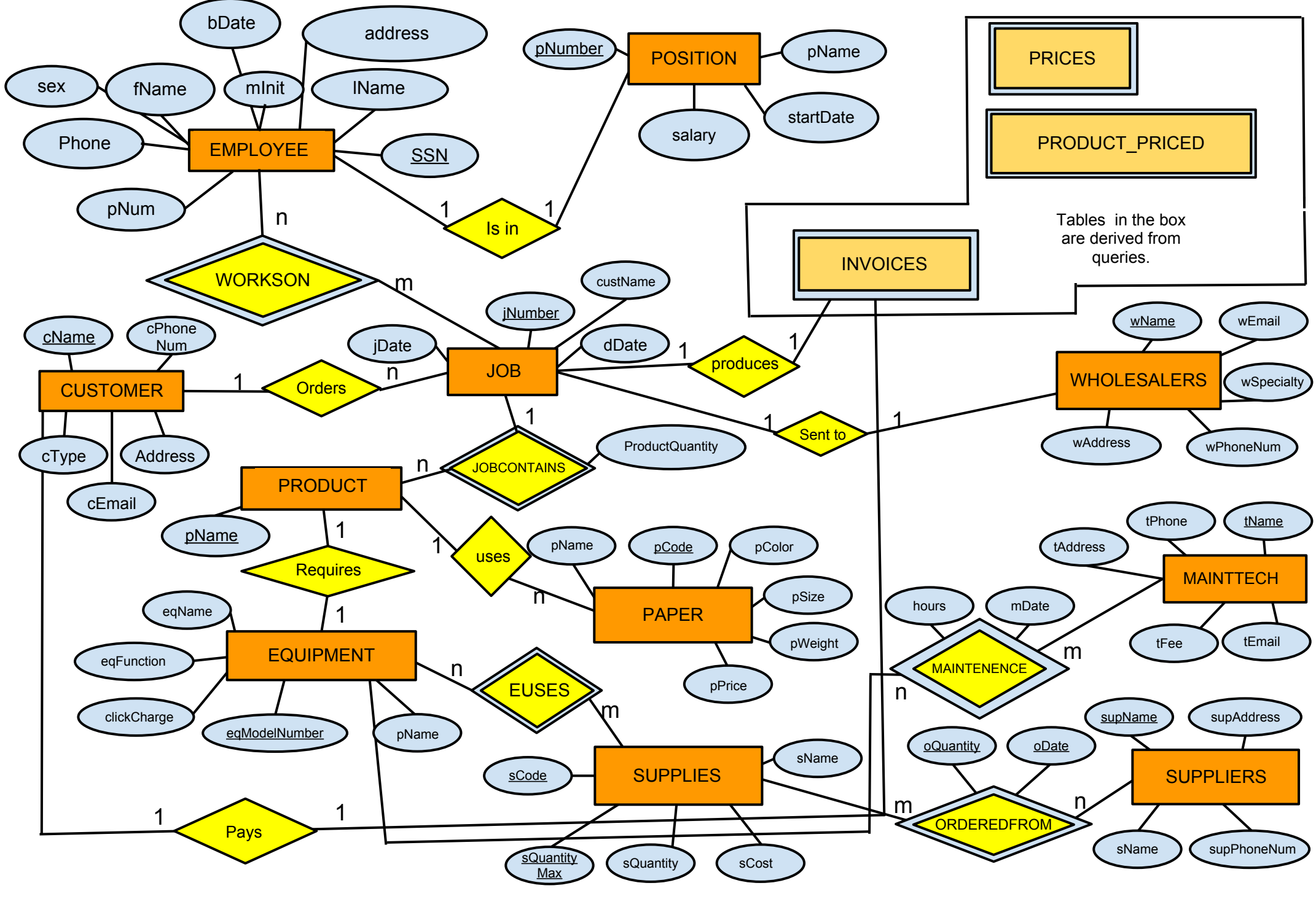
CS 355 Project, 1: Printing Business
Team Members: Lowell Olson, Micah Gemmell, Mark Griffith

The application described in this document is meant to be used in a local printing business. The application will ultimately keep track of the employee's hours, pay, and seniority. It will also keep track of customer orders, mailing address, phone numbers, Projects, consumables, suppliers, equipment, maintenance, products, and services provided by the printing business. The information will be broken down in the following ways.

Employees: Have names, addresses, phone numbers, social security numbers, birthdays, gender and position numbers.
Employees are hired into a position one employee can occupy one position.
Employees work on jobs.
Positions: Have position numbers, names (related to the job), start dates, and base salary.
Customers: have names/business names, Billing Contact name, email, addresses, and phone numbers.
Customers order jobs.
Jobs: Have a number, date, delivery date, and a customer name.
Jobs contain products and their quantities.
Products have a name based on the equipment that creates them, a paper name, paper color and a price.
Prices: are generated based on the click charge from equipment and the paper price.
Paper: has a name, unique code, color, weight, size, and price per sheet. White paper is so inexpensive that it is provided free of charge. This is sometimes used when pricing out jobs that don't have a paper cost (for instance printing on customer stock or using a machine that does not require paper)
Invoices: Are generated from Jobs the products they contain and the prices of those products. They contain the subtotal, tax and total for the job. The job number becomes the invoice number. Invoices are due within 30 days of job delivery.
Supplies: Have a supplier, name, stock number (or other identifying product number, and a manufacturer, an on hand quantity and a maximum quantity.
Consumables are used by the equipment and ordered from suppliers.
Equipment: has a type, a unique ID number, a click charge, a function and a technician.
Equipment: uses supplies.
Technicians: maintain the equipment and are paid based on the maintenance contract. They have a telephone number , email, and hourly fee.
Maintenance: is tracked by the Technician, the date they came in and how many hours they worked on equipment.
Suppliers: Supply consumables used to produce products and each has a unique name, sales phone number and email They also have a product listing with all the consumables they can provide.
The printing business provides products and services in two ways.
In house: These products are produced on site using the equipment and paper, and are based on the customer's order.

Wholesalers: These products are produced off site through a wholesaler. Either the product can-not be produced in house or the order is too large for the in house staff. Wholesalers have unique names, products, phone numbers, and addresses.


EMPLOYEE(fName, mInit, lName, phone, ssn, bDate, address, sex, pNum)
POSITION(pNumber, pName, startDate, sSalary)
JOB(jNumber, jDate, dDate, custName)
WORKSON(jNum, essn)
CUSTOMER(cName, cType, cPhoneNum, cAddress, cEmail)
JOBCONTAINS(productQuantity, jobNum, productName, paperName, paperColor)
PRODUCT(prodName)
PAPER(pName, pCode, pColor, pSize, pWeight, pPrice)
EQUIPMENT(eqName, eqFunction, clickCharge, eqNodelNumber, tech)
SUPPLIES(sName, sCost, sQuantity, sCode)
SUPPLIERS(supName, supAddress, supPhoneNum, supEmail)
WHOLESALERS(wName, wAddress, wPhoneNum, wEmail, wSpecialty)
ORDEREDFROM(spCode, spName, oDate,oQuantity, sPrice)
EUSES(supCode, eModel)
MAINTENANCE(hours, mDate, techName)
MAINTTECH(tName, tPhone, tAddress, tFee, tEmail)


bDate
address
sex
fName
mInit
lName
Phone
EMPLOYEE
SSN
pNum
n
1
Is in
1
pNumber
POSITION
pName
salary
startDate
PRICES
PRODUCT_PRICED
Tables in the box are derived from queries.
INVOICES
WORKSON
m
custName
jNumber
jDate
dDate
cName
cPhone Num
CUSTOMER
1
Orders
n
JOB
1
produces
1
wName
wEmail
WHOLESALERS
wSpecialty
wAddress
wPhoneNum
1
Sent to
1
cType
Address
cEmail
n
JOBCONTAINS
ProductQuantity
PRODUCT
pName
1
Requires
1
1
uses
n
pName
pCode
pColor
PAPER
pSize
pWeight
pPrice
tPhone
tName
tAddress
MAINTTECH
tFee
tEmail
hours
mDate
MAINTENENCE
m
n
eqName
EQUIPMENT
eqFunction
clickCharge
eqModelNumber
pName
n
EUSES
m
sCode
SUPPLIES
sName
sQuantity Max
sQuantity
sCost
supName
supAddress
oQuantity
oDate
SUPPLIERS
m
ORDEREDFROM
n
sName
supPhoneNum
1
Pays
1

SQL Scripts

Primary SQL Script

This script generates our main tables.

```
use CopyCorner;

BEGIN;

set @@foreign_key_checks = 0;


DROP TABLE IF EXISTS employee;


CREATE TABLE employee (
    fName           varchar(20)  NOT NULL,
    mInit           varchar(1),
    lName           varchar(40)  NOT NULL,
    phone           varchar(10),
    ssn     varchar(9)    NOT NULL PRIMARY KEY,
    bDate           date          NOT NULL,
    address         varchar(50)  NOT NULL,
    sex     varchar(1)    NOT NULL,
    pNum            integer                 NOT NULL UNIQUE
);


DROP TABLE IF EXISTS position;


CREATE TABLE position (
```

```
    pNumber         integer(6)     NOT NULL PRIMARY KEY,
    pName           varchar(20)    NOT NULL,
    startDate       date           NOT NULL,
    sSalary         double precision       NOT NULL
);

ALTER TABLE employee ADD CONSTRAINT employee_pNUM_refs_position_pNumber     FOREIGN\
KEY (pNUM) REFERENCES position (pNumber);

DROP TABLE IF EXISTS customer;

CREATE TABLE customer (
    cName           varchar(40)    NOT NULL PRIMARY KEY,
    cType           varchar(1)     NOT NULL,
    cPhoneNum       varchar(10)    NOT NULL UNIQUE,
    cAddress        varchar(50)    NOT NULL,
    cEmail          varchar(50)
);

DROP TABLE IF EXISTS job;

CREATE TABLE job (
    jNumber             integer         NOT NULL PRIMARY KEY,
    jDate               date            NOT NULL,
    dDate               date            NOT NULL,
```

```
    custName            varchar(40)     NOT NULL
);

ALTER TABLE job ADD CONSTRAINT job_custName_refs_customer_cName FOREIGN KEY (custName)
REFERENCES customer (cName);

DROP TABLE IF EXISTS equipment;

CREATE TABLE equipment (
    eqName          varchar(20) NOT NULL UNIQUE,
    eqFunction      varchar(30)  NOT NULL,
    clickCharge     double precision      NOT NULL,
    eqModelNumber   varchar(30)      NOT NULL PRIMARY KEY,
    tech            varchar(40) NOT NULL
);

DROP TABLE IF EXISTS paper;

CREATE TABLE paper (
    pName       varchar(40)         NOT NULL,
    pCode       varchar(40)         NOT NULL PRIMARY KEY,
    pColor      varchar(40)         NOT NULL,
    pWeight     integer             NOT NULL,
    pSize       varchar(10)         NOT NULL,
    pPrice      double precision    NOT NULL
);
```

```
DROP TABLE IF EXISTS product;

CREATE TABLE product (

    prodName      varchar(40)   NOT NULL PRIMARY KEY

);

DROP TABLE IF EXISTS workson;

CREATE TABLE workson (
    jNum          integer             NOT NULL,
    essn          varchar(9)          NOT NULL,

    PRIMARY KEY(essn, jNum)

);

ALTER TABLE workson ADD CONSTRAINT workson_essn_refs_employee_ssn FOREIGN KEY (essn)
REFERENCES employee (ssn);

ALTER TABLE workson ADD CONSTRAINT workson_jNum_refs_job_jNumber FOREIGN KEY (jNum)
REFERENCES job (jNumber);

DROP TABLE IF EXISTS jobcontains;

CREATE TABLE jobcontains (
```

```
    productQuantity     integer NOT NULL,
    jobNum              integer NOT NULL,
    productName         varchar(40)     NOT NULL,
    paperName  varchar(40)      NOT NULL,
    paperColor   varchar(40)      NOT NULL,

    PRIMARY KEY(jobNum, productName)
);
ALTER TABLE jobcontains ADD CONSTRAINT jobcontains_jobNum_refs_job_jNumber FOREIGN KEY (jobNum) REFERENCES job (jNumber);
ALTER TABLE jobcontains ADD CONSTRAINT jobcontains_productName_refs_product_prodName FOREIGN KEY (productName) REFERENCES product (prodName);
ALTER TABLE jobcontains ADD CONSTRAINT jobcontains_paperName_refs_paper_pName FOREIGN KEY (paperName) REFERENCES paper (pName);
ALTER TABLE jobcontains ADD CONSTRAINT jobcontains_paperColor_refs_paper_pColor FOREIGN KEY (paperColor) REFERENCES paper (pColor);

DROP TABLE IF EXISTS supplies;

CREATE TABLE supplies (

    sName           varchar(30)  NOT NULL,
    sQuantity     integer           NOT NULL,
    sQuantityMax   integer                NOT NULL,
```

```
    sCode           varchar(40)  NOT NULL PRIMARY KEY,
    sSupName        varchar(40)  NOT NULL
);

ALTER TABLE supplies ADD CONSTRAINT supplies_sSupName_refs_suppliers_supname FOREIGN KEY
(sSupName) REFERENCES suppliers (supName);

DROP TABLE IF EXISTS suppliers;

CREATE TABLE suppliers (
    supName         varchar(40) NOT NULL PRIMARY KEY,
    supAddress      varchar(50)   NOT NULL,
    supPhoneNum     varchar(10)          NOT NULL,
    supEmail        varchar(40) NOT NULL UNIQUE
);

DROP TABLE IF EXISTS wholesalers;

CREATE TABLE wholesalers (
    wName       varchar(40)     NOT NULL PRIMARY KEY,
    wAddress    varchar(50)     NOT NULL,
    wPhoneNum   varchar(10)     NOT NULL,
    wEmail      varchar(40)     NOT NULL UNIQUE,
    wSpecialty  varchar(40)     NOT NULL
);
```

```
DROP TABLE IF EXISTS orderedfrom;


CREATE TABLE orderedfrom (
    spCode              varchar(40)     NOT NULL,
    spName              varchar(40)     NOT NULL,
    oDate               date,
    oQuantity   integer,
    sPrice              double precision,


    PRIMARY KEY(spCode, spName)
);


ALTER TABLE orderedfrom ADD CONSTRAINT orderedfrom_spCode_refs_supplies_sCode FOREIGN KEY
(spCode) REFERENCES supplies (sCode);

ALTER TABLE orderedfrom ADD CONSTRAINT orderedfrom_spName_refs_suppliers_supName FOREIGN
KEY (spName) REFERENCES suppliers (supName);

ALTER TABLE orderedfrom ADD CONSTRAINT orderedfrom_sPrice_refs_supplies_sCost FOREIGN KEY
(sPrice) REFERENCES supplies (sCost);




DROP TABLE IF EXISTS euses;


CREATE TABLE euses (
    supCode     varchar(40)             NOT NULL,
    eModel      varchar(30)             NOT NULL,
```

```
    PRIMARY KEY(supCode, eModel)
);

ALTER TABLE euses ADD CONSTRAINT euses_supCode_refs_supplies_sCode FOREIGN KEY (supCode)
REFERENCES supplies (sCode);

ALTER TABLE euses ADD CONSTRAINT euses_eModel_refs_equipment_eqModelNumber FOREIGN KEY
(eModel) REFERENCES equipment (eqModelNumber);

DROP TABLE IF EXISTS maintenance;

CREATE TABLE maintenance (

    hours           integer NOT NULL,
    mDate           date        NOT NULL,
    techName        varchar(40)     NOT NULL,

    PRIMARY KEY (techName, mDate)
);

ALTER TABLE maintenance ADD CONSTRAINT maintenance_techName_refs_mainttech_tName FOREIGN
KEY (techName) REFERENCES mainttech (tName);

DROP TABLE IF EXISTS mainttech;

CREATE TABLE mainttech (
    tName       varchar(40)         NOT NULL PRIMARY KEY,
    tPhone      varchar(10)         NOT NULL,
    tAddress    varchar(50)         NOT NULL,
```

```
    tFee            integer                 NOT NULL,
    tEmail          varchar(40)
);
ALTER TABLE equipment ADD CONSTRAINT equipment_tech_refs_mainttech_tName FOREIGN KEY
(tech) REFERENCES mainttech (tName);



set @@foreign_key_checks = 1;


COMMIT;
```

Additional Script File

This script runs after the initial table creating script runs in order to generate our 3 tables created from queries, our 3 views, and our 6 indices.

```
use CopyCorner;

BEGIN;

set @@foreign_key_checks = 0;


DROP TABLE IF EXISTS product_priced;


CREATE TABLE product_priced


select prodName as Product_Name, pName as Paper_Name, pColor as Paper_Color, pCode, (pPrice + clickCharge) as Price

from product, paper, equipment

where prodName = eqFunction and (prodName != 'BW Copy' or prodName != 'Color Copy' or prodName != 'Press Printing') and pCode = '0000-0000'

UNION

select prodName as Product_Name, pName as Paper_Name, pColor as Paper_Color, pCode, (pPrice + clickCharge) as Price

from product, paper, equipment

where prodName = eqFunction and (prodName = 'BW Copy' or prodName = 'Color Copy' or prodName = 'Press Printing');


DROP TABLE IF EXISTS prices;
```

```
CREATE TABLE prices


select jobNum, productName, paperName, paperColor, productQuantity as Number_of_Units, Price as Price_Per_Unit, (Price * productQuantity) as productSubtotal

from jobcontains, job, product_priced

where jobNum = jNumber and productName = Product_Name and paperName = Paper_Name and Paper_Color = paperColor;


DROP TABLE IF EXISTS invoices;


CREATE TABLE invoices


select jnumber as Invoice_Number, jDate as 'Date', date_add(dDate, INTERVAL 30 DAY) as Remit_by, cName as 'Name', cAddress, sum(productSubtotal) as Subtotal, (sum(productSubtotal)*.08) as Tax, (sum(productSubtotal)+(sum(productSubtotal)*.08))as Total

from job, customer, prices

where cName = custName and jNumber = jobNum

group by jnumber;

Alter Table invoices ADD Paid VARCHAR(10);

update invoices

set Paid = 'FALSE';

update invoices

set Paid = 'TRUE' where Invoice_Number = 21801 or Invoice_Number = 22003 or Invoice_Number = 22801 or Invoice_Number = 120310;


DROP VIEW IF EXISTS late_invoices;
```

```
CREATE VIEW late_invoices as

select Invoice_Number, Name, Remit_by, datediff(curdate(),Remit_by) as Days_Overdue, cPhoneNum as Phone_Number

from invoices, customer

where datediff(curdate(), Remit_by) >0 and cName = Name;


DROP VIEW IF EXISTS jobs_per_month;


Create View jobs_per_month as

select monthname(date) as 'Month', year(date) as 'Year', count(*) as 'Number ofJobs'from invoices group by extract(month from date);


DROP VIEW IF EXISTS low_supplies;


create view low_supplies as

select sName as Supply_Name, sQuantity as Amount_Left, sSupName as 'Supplier', supPhoneNum as Phone_Number

from supplies, suppliers where (sQuantity <= (sQuantityMax *.1)) and supName = sSupName ;


CREATE INDEX tech_name_idx ON mainttech (tName);

CREATE INDEX emp_ssn_position_idx ON employee (ssn, pNum);

CREATE INDEX customer_name_idx ON customer (cName);

CREATE INDEX paper_name_paper_color_idx ON product_priced (Paper_Name, Paper_Color);

CREATE INDEX invoice_numbers_idx ON invoices (Invoice_Number);

CREATE INDEX paper_codes_idx ON paper (pCode);
```

```
set @@foreign_key_checks = 1;


COMMIT;
```

Hardcoded Tables

Tables

Table 1: Customer

This table lists customers' names, type (Individual or Business), phone number, address and email address.

cName	cType	cPhoneNum	cAddress	cEmail
John Adams	I	7073457234	123 Technology Way, Napa, CA	jon332@yahoo.com
Econo Foods	B	3342567234	444 Food Court, Sonoma, CA	econofood@gmail.com
Joes Gas N Go	B	7076673462	73 Fuel Stop Ave, American Canyon, CA	gas@conoco.com
Bill Williams	I	7772258345	7723 Winding Road, SaintHelena, CA	Bwilliams@gmail.com
Dan Tucker	I	3356720918	1171 Hooten Holler, Hicksville, CA	NULL
Jeff Jepson	I	2267359090	1412 Anderson Ct, Pleasonton, AL	jj@yahoo.com
Sunny Hill	B	7074563523	12 Jones Street, Fairfield, CA	SH@Gmail.com
BBQ City	B	7079942373	222 Lincoln Way, Jonesborough, AR	BBQC@ether.net
Shady Lane	B	3332474534	100 Oak Drive, Peterborough, UK	SLHome@co.uk.net

Table 2: Employee

This table lists employees' names, phone numbers, social security numbers, birthdates, addresses, sex, position number.

fName	mInit	lName	phone	ssn	bDate	address	sex	pNum
John	B	Smith	7071234567	123456789	1970-10-08	37 Yemen Ct. Yemen, Yountville, CA	M	123456
Dave	J	Thomas	7075439090	234567890	1934-03-10	134 Aman Dr., Petaluma, CA	M	234567
Anne	D	Sanders	7074329086	345678901	1990-10-12	222 Jones Circle, Fairfield, CA	F	345678
Betty	R	Johnson	3342319078	456789012	1963-04-17	55 Trujillo Way, SantaAna, NM	F	456789
James	K	Polk	6783451234	567890123	1857-01-03	7117 Apt 2, Janice Ave, Burnsville, MN	M	567890
Johnny	B	Goode	7078783344	678901234	1987-07-03	323 Apt 47, Janice Ave, Burnsville, MN	F	678901
Steve	V	Wonder	8323450098	789012345	1977-11-22	768 Stevens Ct., Fayetteville, NC	M	789012
Andy	R	Taylor	2342234980	890123456	1967-03-19	222 Waterview Way, Johnson, CA	M	890123

Table 3: Equipment

This table stores each piece of equipment's Name, Function (the service it performs), Click Charge (the price per use), Model Number and the name of the technician that maintains it.

eqName	eqFunction	clickCharge	eqModelNumber	tech
Docucolor 260	Color Copy	0.49	DRB132502	Mark Rockroth
Rhin-o-tuff Binder	Spiral Binding	2	OD4012	Scott Ellsworth
Heidelberg Printmast	Press Printing	0.04	962817	Eric Blanc
Konica Bizhub 1050	BW Copy	0.08	56UE00347	Joel Quantino
714 UltraFold	Folding	0.01	B714XLT	Scott Ellsworth
Rosback Scorer	Scoring	0.01	2207615683	Scott Ellsworth

Table 4: Euses

This table tracks by code which piece of equipment uses which supplies.

supCode	eModel
BTO	56UE00347
BTO	DRB132502
CTO	DRB132502
MTO	DRB132502
SPB	OD4012
YTO	DRB132502

Table 5: Job

This table tracks a a Job's Job Number, Job Date, Delivery Date and name of the Customer who ordered it.

jNumber	jDate	dDate	custName
22801	2012-02-28	2013-10-03	John Adams
22101	2013-02-21	2013-03-02	Econo Foods
13103	2013-01-31	2013-02-05	Joes Gas N Go
22402	2013-02-15	2013-03-13	Dan Tucker
12002	2013-01-20	2013-02-03	Econo Foods
22003	2013-02-03	2013-03-02	Econo Foods
11901	2013-01-19	2013-02-03	Econo Foods
22102	2013-02-21	2013-04-02	Joes Gas N Go
12502	2013-01-02	2013-01-15	Joes Gas N Go
21801	2013-02-18	2013-04-01	Joes Gas N Go
120310	2012-02-13	2013-02-10	Jeff Jepson
12910	2012-01-29	2012-02-10	ShadyLane

Table 6: Job Contains

This table contains the quantity and composition of each product ordered as part of a job.

productQuantity	jobNum	productName	paperName	paperColor
10	22801	BW Copy	Classic Crest Text	Avalanche White
100	22101	Color Copy	Classic Crest Cover	Avon Brilliant White
500	13103	Press Printing	Classic Crest Text	Solar White
200	22402	Folding	Plain Paper	White
200	22402	Scoring	Plain Paper	White
400	12002	Color Copy	Classic Crest Writing	Baronial Ivory
40	12002	Spiral Binding	Plain Paper	White
1000	22003	BW Copy	Plain Paper	White
300	11901	BW Copy	Classic Crest Text	Avalanche White
5000	22102	Color Copy	Classic Crest Cover	Solar White
500	22102	Folding	Plain Paper	White
3000	12502	Scoring	Plain Paper	White
300	12502	Press Printing	Plain Paper	White
200	21801	BW Copy	Classic Crest Writing	Potomac Blue
3000	120310	Spiral Binding	Plain Paper	White
900	12910	BW Copy	Classic Crest Cover	Classic Natural White
90	12910	Folding	Plain Paper	White

Table 7: Maintenance

This table tracks the dates specific Maintenance Techs serviced equipment and the number of hours they spent doing so.

hours	mDate	techName
3	2012-10-08	Mark Rockroth
1	2011-09-14	Joel Quantino
5	2013-02-14	Scott Ellsworth
4	2011-01-31	Eric Blanc
2	2010-06-12	Joel Quantino
6	2011-04-13	Scott Ellsworth
10	2010-09-13	Eric Blanc
8	2011-07-04	Mark Rockroth
3	2013-01-29	Scott Ellsworth
7	2009-12-21	Joel Quantino
11	2012-11-07	Mark Rockroth
16	2011-03-21	Eric Blanc
8	2012-10-09	Mark Rockroth
16	2012-05-22	Scott Ellsworth

5	2012-07-20	Joel Quantino
9	2011-06-21	Eric Blanc
13	2011-03-20	Mark Rockroth
9	2012-01-10	Eric Blanc
6	2013-03-02	Scott Ellsworth
9	2013-02-10	Mark Rockroth

Table 8: Mainttech

This table contains the names, fees and contact information for each of the Maintenance Technicians.

tName	tPhone	tAddress	tFee	tEmail
Mark Rockroth	7078975463	254 H Street, Napa, CA	45	rockerroth@gmail.com
Joel Quantino	7072234587	35 Lincoln Avenue, Vallejo, CA	60	superjoel@yahoo.com
Scott Ellsworth	4156183245	3645 Cabot Boulevard, Emeryville, CA	30	scottnray@att.net
Eric Blanc	7079628765	456 Washington Drive, American Canyon, CA	40	blanccheck@att.net

Table 9: Ordered From

This table contains records of when/where we ordered supplies from, the quantity ordered, and the price per.

spCode	spName	oDate	oQuantity	sPrice
BTO	Deep Toner	2012-12-31	2000	0.01
MTO	Deep Toner	2012-12-31	1000	0.05
CTO	Deep Toner	2012-12-10	1000	0.05
YTO	Officemax	2013-02-14	1500	0.05
SPB	Bindings are us	2013-01-27	100	0.5

Table 10: Paper

This table contains a list of Papers and their various attributes and prices. Make note that plain paper is so inexpensive that it is is no extra cost to a customer in order to keep prices constant for when they bring their own paper in or for calculating machine charges where paper simply is not consumed.

pName	pCode	pColor	pWeight	pSize	pPrice
Classic Crest Text	0145-0229A	Avalanche White	70	8.5x11	0.07
Classic Crest Text	0145-0229B	Solar White	70	8.5x11	0.07
Classic Crest Text	0145-0237A	Avon Brilliant White	80	8.5x11	0.07
Classic Crest Text	0145-0237B	Classic Natural White	80	8.5x11	0.07
Classic Crest Writing	0145-0205A	Antique Gray	24	8.5x11	0.06
Classic Crest Writing	0145-0205B	Avalanche White	24	8.5x11	0.06
Classic Crest Writing	0145-0201A	Avon Brillian White	24	8.5x11	0.06
Classic Crest Writing	0145-0205C	Baronial Ivory	24	8.5x11	0.06

Classic Crest Writing	0145-0205D	Classical Cream	24	8.5x11	0.06
Classic Crest Writing	0145-0201B	Classic Natural White	24	8.5x11	0.06
Classic Crest Writing	0145-0213A	Earthstone	24	8.5x11	0.06
Classic Crest Writing	0145-0213B	Millstone	24	8.5x11	0.06
Classic Crest Writing	0145-0205G	Potomac Blue	24	8.5x11	0.06
Classic Crest Cover	0145-0245A	Avon Brilliant White	65	8.5x11	0.15
Classic Crest Cover	0145-0245B	Classic Natural White	65	8.5x11	0.15
Classic Crest Cover	0145-0248H	Solar White	65	8.5x11	0.15
Classic Crest Cover	0145-0261A	Antique Gray	80	8.5x11	0.18
Classic Crest Cover	0145-0261B	Avalanche White	80	8.5x11	0.18
Classic Crest Cover	0145-0253A	Avon Brilliant White	80	8.5x11	0.18
Classic Crest Cover	0145-0261C	Baronial Ivory	80	8.5x11	0.18
Classic Crest Cover	0145-0272N	Canyon Brown	80	8.5x11	0.24
Classic Crest Cover	0145-0261D	Classic Cream	80	8.5x11	0.18
Classic Crest Cover	0145-0253B	Classic Natural White	80	8.5x11	0.18
Classic Crest Cover	0145-0269A	Earthstone	80	8.5x11	0.19
Classic Crest Cover	0145-0272L	Epic Black	80	8.5x11	0.24
Plain Paper	0000-0000	White	24	8.5x11	0

Table 11: Position

This table tracks the employee Position Numbers, Names, Start Dates, Salaries and Bonuses.

pNumber	pName	startDate	sSalary	bonus
123456	Printer	2013-02-28	30000	0
234567	Typesetter	2012-03-13	33000	0
345678	Manager	1999-05-15	43000	37
456789	Copy Tech	2012-08-11	25000	68
567890	Copy Tech 2	2013-01-20	30000	66
678901	Trainee	2013-02-27	20000	0
789012	Trainee	2013-03-01	20000	0
890123	Copy Tech	2012-09-22	25000	0

Table 12: Product

This table merely contains product names but is necessary for joining together with query tables.

prodName
BW Copy
Color Copy
Folding
Press Printing
Scoring
Spiral Binding

Table 13: Suppliers

This table tracks the Supply Suppliers' Names and contact information

supName	supAddress	supPhoneNum	supEmail
JC Paper	1001 Waterborough Way, Pittsburg, PA	4237860010	jc@jcpaper.com
Deep Toner	2345 Chemical Circle, Newark, NJ	7732340876	tonedown@gmail.com
Bindings are us	14 Binding ave, Lincolon, NE	6632319367	bindertown@aaa.net
Officemax	372 Hanover street, Napa, CA	7079434536	Omax@office.net
Office Depot	299 Air Base Parkway, Fairfield, CA	7074562364	Depottime@world.net

Table 14: Supplies

This table tracks Supplies' Names, on-hand quantities, maximum quantities, codes, and supplier name.

sName	sQuantity	sQuantityMax	sCode	sSupName
BLACK Toner	6300	15000	BTO	Deep Toner
MAGENTA Toner	4500	5000	MTO	Deep Toner
CYAN Toner	4000	5000	CTO	Deep Toner
YELLOW Toner	100	5000	YTO	Deep Toner
BDR spirals	25	500	SPB	Bindings are us

Table 15: Wholesalers

NOTE: This table did not get used in this project, we merely decided to keep it in case we decided to use it for the next part of the project. It tracks information regarding the Names, Contact information and Specialties of the wholesalers the Copy Corner uses.

wName	wAddress	wPhoneNum	wEmail	wSpecialty
4 Over	1001 Colombus St., Glendale, CA	8437210010	4_Over.com	Full Color
Thermcraft	2345 A Street, Sacramento, CA	7073451232	thermcraft@gmail.com	Business Cards
Discount Labels	14 Indiana Rd., Indeanapolis, IN	3412343367	DL@aaa.net	Labels
Spectra	372 Hanover ave, Folsom, CA	7238434536	Spectra@spark.net	Stamps
Imperial Die	323 Seafront Parkway, Hayward, CA	7073244364	Imperialdie@world.net	Foiling

Table 16: Works On

This table tracks which employee works on which job.

jNum	essn
11901	123456789
22801	123456789
11901	234567890
22801	234567890
12502	345678901
13103	345678901
22101	345678901
22102	345678901
12002	456789012
21801	456789012
22402	456789012
120310	456789012
22003	567890123
120310	567890123

Tables Created From Queries

Tables Created From Queries

Table 1: product_priced

This table is one of Mark's personal favorites. This table is created from a union of two queries that takes product names from the product table, equipment click charges from the equipment table and paper names, colors and prices from the paper table. From this union it creates a permutation of every possible combination of equipment and paper and calculates the price per. This also takes into account that some machines do not consume paper; these machines are assigned only to "Plain Paper" because it has a price of 0.00 and thus will not throw off the price.

```
create table product_priced

select prodName as Product_Name, pName as Paper_Name, pColor as Paper_Color, pCode, (pPrice + clickCharge) as Price
from product, paper, equipment
where prodName = eqFunction and (prodName != 'BW Copy' or prodName != 'Color Copy' or prodName != 'Press Printing') and pCode = '0000-0000'

UNION

select prodName as Product_Name, pName as Paper_Name, pColor as Paper_Color, pCode, (pPrice + clickCharge) as Price
from product, paper, equipment
where prodName = eqFunction and (prodName = 'BW Copy' or prodName = 'Color Copy' or prodName = 'Press Printing');
```

Product_Name	Paper_Name	Paper_Color	pCode	Price
BW Copy	Plain Paper	White	0000-0000	0.08
BW Copy	Classic Crest Text	Avalanche White	0145-0229A	0.15
BW Copy	Classic Crest Text	Solar White	0145-0229B	0.15
BW Copy	Classic Crest Text	Avon Brilliant White	0145-0237A	0.15
BW Copy	Classic Crest Text	Classic Natural White	0145-0237B	0.15
BW Copy	Classic Crest Writing	Antique Gray	0145-0205A	0.14
BW Copy	Classic Crest Writing	Avalanche White	0145-0205B	0.14
BW Copy	Classic Crest Writing	Avon Brillian White	0145-0201A	0.14
BW Copy	Classic Crest Writing	Baronial Ivory	0145-0205C	0.14
BW Copy	Classic Crest Writing	Classical Cream	0145-0205D	0.14
BW Copy	Classic Crest Writing	Classic Natural White	0145-0201B	0.14
BW Copy	Classic Crest Writing	Earthstone	0145-0213A	0.14
BW Copy	Classic Crest Writing	Millstone	0145-0213B	0.14
BW Copy	Classic Crest Writing	Potomac Blue	0145-0205G	0.14
BW Copy	Classic Crest Cover	Avon Brilliant White	0145-0245A	0.23
BW Copy	Classic Crest Cover	Classic Natural White	0145-0245B	0.23
BW Copy	Classic Crest Cover	Solar White	0145-0248H	0.23
BW Copy	Classic Crest Cover	Antique Gray	0145-0261A	0.26
BW Copy	Classic Crest Cover	Avalanche White	0145-0261B	0.26
BW Copy	Classic Crest Cover	Avon Brilliant White	0145-0253A	0.26
BW Copy	Classic Crest Cover	Baronial Ivory	0145-0261C	0.26
BW Copy	Classic Crest Cover	Canyon Brown	0145-0272N	0.32

	BW Copy	Classic Crest Cover	Classic Cream	0145-0261D	0.26
	BW Copy	Classic Crest Cover	Classic Natural White	0145-0253B	0.26
	BW Copy	Classic Crest Cover	Earthstone	0145-0269A	0.27
	BW Copy	Classic Crest Cover	Epic Black	0145-0272L	0.32
	Color Copy	Plain Paper	White	0000-0000	0.49
	Color Copy	Classic Crest Text	Avalanche White	0145-0229A	0.56
	Color Copy	Classic Crest Text	Solar White	0145-0229B	0.56
	Color Copy	Classic Crest Text	Avon Brilliant White	0145-0237A	0.56
	Color Copy	Classic Crest Text	Classic Natural White	0145-0237B	0.56
	Color Copy	Classic Crest Writing	Antique Gray	0145-0205A	0.55
	Color Copy	Classic Crest Writing	Avalanche White	0145-0205B	0.55
	Color Copy	Classic Crest Writing	Avon Brillian White	0145-0201A	0.55
	Color Copy	Classic Crest Writing	Baronial Ivory	0145-0205C	0.55
	Color Copy	Classic Crest Writing	Classical Cream	0145-0205D	0.55
	Color Copy	Classic Crest Writing	Classic Natural White	0145-0201B	0.55
	Color Copy	Classic Crest Writing	Earthstone	0145-0213A	0.55
	Color Copy	Classic Crest Writing	Millstone	0145-0213B	0.55
	Color Copy	Classic Crest Writing	Potomac Blue	0145-0205G	0.55
	Color Copy	Classic Crest Cover	Avon Brilliant White	0145-0245A	0.64
	Color Copy	Classic Crest Cover	Classic Natural White	0145-0245B	0.64
	Color Copy	Classic Crest Cover	Solar White	0145-0248H	0.64
	Color Copy	Classic Crest Cover	Antique Gray	0145-0261A	0.67
	Color Copy	Classic Crest Cover	Avalanche White	0145-0261B	0.67
	Color Copy	Classic Crest Cover	Avon Brilliant White	0145-0253A	0.67
	Color Copy	Classic Crest Cover	Baronial Ivory	0145-0261C	0.67
	Color Copy	Classic Crest Cover	Canyon Brown	0145-0272N	0.73
	Color Copy	Classic Crest Cover	Classic Cream	0145-0261D	0.67
	Color Copy	Classic Crest Cover	Classic Natural White	0145-0253B	0.67
	Color Copy	Classic Crest Cover	Earthstone	0145-0269A	0.68
	Color Copy	Classic Crest Cover	Epic Black	0145-0272L	0.73
	Folding	Plain Paper	White	0000-0000	0.01
	Press Printing	Plain Paper	White	0000-0000	0.04
	Press Printing	Classic Crest Text	Avalanche White	0145-0229A	0.11
	Press Printing	Classic Crest Text	Solar White	0145-0229B	0.11
	Press Printing	Classic Crest Text	Avon Brilliant White	0145-0237A	0.11
	Press Printing	Classic Crest Text	Classic Natural White	0145-0237B	0.11
	Press Printing	Classic Crest Writing	Antique Gray	0145-0205A	0.1
	Press Printing	Classic Crest Writing	Avalanche White	0145-0205B	0.1
	Press Printing	Classic Crest Writing	Avon Brillian White	0145-0201A	0.1
	Press Printing	Classic Crest Writing	Baronial Ivory	0145-0205C	0.1
	Press Printing	Classic Crest Writing	Classical Cream	0145-0205D	0.1
	Press Printing	Classic Crest Writing	Classic Natural White	0145-0201B	0.1
	Press Printing	Classic Crest Writing	Earthstone	0145-0213A	0.1
	Press Printing	Classic Crest Writing	Millstone	0145-0213B	0.1
	Press Printing	Classic Crest Writing	Potomac Blue	0145-0205G	0.1
	Press Printing	Classic Crest Cover	Avon Brilliant White	0145-0245A	0.19
	Press Printing	Classic Crest Cover	Classic Natural White	0145-0245B	0.19
	Press Printing	Classic Crest Cover	Solar White	0145-0248H	0.19
	Press Printing	Classic Crest Cover	Antique Gray	0145-0261A	0.22
	Press Printing	Classic Crest Cover	Avalanche White	0145-0261B	0.22
	Press Printing	Classic Crest Cover	Avon Brilliant White	0145-0253A	0.22

Press Printing	Classic Crest Cover	Baronial Ivory	0145-0261C	0.22
Press Printing	Classic Crest Cover	Canyon Brown	0145-0272N	0.28
Press Printing	Classic Crest Cover	Classic Cream	0145-0261D	0.22
Press Printing	Classic Crest Cover	Classic Natural White	0145-0253B	0.22
Press Printing	Classic Crest Cover	Earthstone	0145-0269A	0.23
Press Printing	Classic Crest Cover	Epic Black	0145-0272L	0.28
Scoring	Plain Paper	White	0000-0000	0.01
Spiral Binding	Plain Paper	White	0000-0000	2

Table 2: Prices

This table takes the information from product_priced and applies it to the items in jobcontains to generate the prices of each product in a job.

create table prices as

select jobNum, productName, paperName, paperColor, productQuantity as Number_of_Units, Price as Price_Per_Unit, (Price * productQuantity) as productSubtotal

from jobcontains, job, product_priced where jobNum = jNumber and productName = Product_Name and paperName = Paper_Name and Paper_Color = paperColor;

jobNum	productName	paperName	paperColor	Number_of_Units	Price_Per_Unit	productSubtotal
11901	BW Copy	Classic Crest Text	Avalanche White	300	0.15	45
12002	Color Copy	Classic Crest Writing	Baronial Ivory	400	0.55	220
12002	Spiral Binding	Plain Paper	White	40	2	80
12502	Press Printing	Plain Paper	White	300	0.04	12
12502	Scoring	Plain Paper	White	3000	0.01	30
12910	BW Copy	Classic Crest Cover	Classic Natural White	900	0.23	207
12910	BW Copy	Classic Crest Cover	Classic Natural White	900	0.26	234
12910	Folding	Plain Paper	White	90	0.01	0.9
13103	Press Printing	Classic Crest Text	Solar White	500	0.11	55
21801	BW Copy	Classic Crest Writing	Potomac Blue	200	0.14	28
22003	BW Copy	Plain Paper	White	1000	0.08	80
22101	Color Copy	Classic Crest Cover	Avon Brilliant White	100	0.64	64
22101	Color Copy	Classic	Avon	100	0.67	67

		Crest Cover	Brilliant White			
22102	Color Copy	Classic Crest Cover	Solar White	5000	0.64	3200
22102	Folding	Plain Paper	White	500	0.01	5
22402	Folding	Plain Paper	White	200	0.01	2
22402	Scoring	Plain Paper	White	200	0.01	2
22801	BW Copy	Classic Crest Text	Avalanche White	10	0.15	1.5
120310	Spiral Binding	Plain Paper	White	3000	2	6000

Table 3: Invoices

This table creates invoices based on the job, customer and prices tables. The job number becomes the invoice number. The date of the remittance is generated by adding 30 days to the delivery date. The customer name and address are pulled in for mailing. The sum of the subtotals in the products in each job is given as an invoice subtotal and then tax and total are calculated off of that. After creation by query the table then has a 'Paid' column added with a default value of 'False'

```
create table invoices as

select jnumber as Invoice_Number, jDate as 'Date', date_add(dDate, INTERVAL 30 DAY) as Remit_by, cName as 'Name',
cAddress as Address, sum(productSubtotal) as Subtotal, (sum(productSubtotal) *.08) as Tax, (sum(productSubtotal) +
(sum(productSubtotal) *.08)) as Total

from job, customer, prices

where cName = custName and jNumber = jobNum

group by jnumber;

Alter Table invoices ADD Paid VARCHAR(10);

update invoices

set Paid = 'False';
```

Invoice_ Number	Date	Remit_by	Name	Address	Subtotal	Tax	Total
11901	2013-01-19	2013-03-05	Econo Foods	444 Food Court, Sonoma, CA	45	3.6	48.6
12002	2013-01-20	2013-03-05	Econo Foods	444 Food Court, Sonoma, CA	300	24	324
12502	2013-01-02	2013-02-14	Joes Gas N Go	73 Fuel Stop Ave, American Canyon, CA	42	3.36	45.36
13103	2013-01-31	2013-03-07	Joes Gas N Go	73 Fuel Stop Ave, American Canyon, CA	55	4.4	59.4
21801	2013-02-18	2013-05-01	Joes Gas N Go	73 Fuel Stop Ave, American Canyon, CA	28	2.24	30.24
22003	2013-02-03	2013-04-01	Econo Foods	444 Food Court, Sonoma, CA	80	6.4	86.4
22101	2013-02-21	2013-04-01	Econo Foods	444 Food Court, Sonoma, CA	131	10.48	141.48
22102	2013-02-21	2013-05-02	Joes Gas N Go	73 Fuel Stop Ave, American Canyon, CA	3205	256.4	3461.4
22402	2013-02-15	2013-04-12	Dan Tucker	1171 Hooten Holler, Hicksville, CA	4	0.32	4.32
22801	2012-02-28	2013-11-02	John Adams	123 Technology Way, Napa, CA	1.5	0.12	1.62
120310	2012-02-13	2013-03-12	Jeff Jepson	1412 Anderson Ct, Pleasonton, AL	6000	480	6480

Non-Trivial Queries

1. We want to get the the name and phone number of a technician who works on the equipment that we are currently having trouble with.

select eqName as "Equipment Name", tName as "Technician Name", tPhone as "Phone" from equipment,mainttech where tName = tech and eqName='Docucolor 260';

```
+---------------+----------------+------------+
| Equipment Name | Technician Name | Phone      |
+---------------+----------------+------------+
| Docucolor 260  | Mark Rockroth   | 7078975463 |
+---------------+----------------+------------+
1 row in set (0.00 sec)
```

2. We want to figure out which customer has spent the most from our company. Eventually this could be used for a valued customer feature.

select max(t), n from (select sum(total) as t, Name as n from invoices group by Name order by total desc) as p;

```
+--------+-------------+
| max(t) | n           |
+--------+-------------+
| 551.88 | Econo Foods |
+--------+-------------+
1 row in set (0.00 sec)
```

3. We want to know how many hours each technician has worked so that we can determine which technicians are the fastest worker.

select sum(hours) as thours, techName, extract(year from mdate)as year, extract(month from mdate) as month from maintenance group by month,year order by mdate asc;

```
+--------+-----------------+------+-------+
| thours | techName        | year | month |
+--------+-----------------+------+-------+
|      7 | Joel Quantino   | 2009 |    12 |
|      2 | Joel Quantino   | 2010 |     6 |
|     10 | Eric Blanc      | 2010 |     9 |
|      4 | Eric Blanc      | 2011 |     1 |
|     29 | Eric Blanc      | 2011 |     3 |
|      6 | Scott Ellsworth | 2011 |     4 |
|      9 | Eric Blanc      | 2011 |     6 |
|      8 | Mark Rockroth   | 2011 |     7 |
|      1 | Joel Quantino   | 2011 |     9 |
|      9 | Eric Blanc      | 2012 |     1 |
|     16 | Scott Ellsworth | 2012 |     5 |
|      5 | Joel Quantino   | 2012 |     7 |
|     11 | Mark Rockroth   | 2012 |    10 |
```

```
|     11 | Mark Rockroth   | 2012 |    11 |
|      3 | Scott Ellsworth | 2013 |     1 |
|     14 | Scott Ellsworth | 2013 |     2 |
|      6 | Scott Ellsworth | 2013 |     3 |
+--------+-----------------+------+-------+
17 rows in set (0.00 sec)
```

4. Find the total amount a person earned.

```
select sum(s) as "total earned", count(c) as "#ofjobs", fname, lname from (select sum(total) as s, count(total) as c, essn from invoices, workson where Invoice_Number=jNum group by Invoice_Number, essn) as m, employee where essn=ssn group by essn;
+--------------+---------+-------+---------+
| total earned | #ofjobs | fname | lname   |
+--------------+---------+-------+---------+
|        50.22 |       2 | John  | Smith   |
|        50.22 |       2 | Dave  | Thomas  |
|      3707.64 |       4 | Anne  | Sanders |
|      6838.56 |       4 | Betty | Johnson |
|       6566.4 |       2 | James | Polk    |
+--------------+---------+-------+---------+
5 rows in set (0.00 sec)
```

5. Find out how many jobs an employee worked on in a month. This is useful to determine the business of our business.

select fname, lname, count(jnum) from (select fname, lname, count(jnum), jnum from employee, workson where ssn=essn group by jnum) as jobs, job where extract(month from job.jDate) = '2' and jnum=jnumber group by lname;

```
+-------+---------+-------------+
| fname | lname   | count(jnum) |
+-------+---------+-------------+
| Betty | Johnson |           3 |
| James | Polk    |           1 |
| Anne  | Sanders |           3 |
| John  | Smith   |           1 |
+-------+---------+-------------+
4 rows in set (0.00 sec)
```

6. Along the same lines of the above query, we want to determine how busy our company was during given month.

```
select CASE
when extract(month from jdate) = 1 then 'January'
when extract(month from jdate) = 2 then 'February'
when extract(month from jdate) = 3 then 'March'
when extract(month from jdate) = 4 then 'April'
when extract(month from jdate) = 5 then 'May'
when extract(month from jdate) = 6 then 'June'
when extract(month from jdate) = 7 then 'July'
when extract(month from jdate) = 8 then 'August'
when extract(month from jdate) = 9 then 'September'
when extract(month from jdate) = 10 then 'October'
when extract(month from jdate) = 11 then 'November'
else 'December' end
as Month, count(jNumber) as "Jobs/Month" from job group by extract(month from jDate);
+----------+------------+
| month    | Jobs/Month |
+----------+------------+
| December |          4 |
| February |          8 |
+----------+------------+
2 rows in set, 1 warning (0.00 sec)
```

7. We want to find out what is the most popular product overall. Then we could use this to up the prices.

```
select max(pq) as "Quantity", pn as "Popular Product" from (select sum(productQuantity) as pq,
productName as pn from jobcontains group by productName order by pq desc) as t;
+----------+-----------------+
| Price of | Popular Product |
+----------+-----------------+
|     5500 | Color Copy      |
+----------+-----------------+
1 row in set (0.00 sec)
```

8. We want to know what our most profitable product is.

select p as "Most Popular Product", max(s) as "Profit" from (select sum(ProductSubtotal) as s, ProductName as p from Prices group by productName order by s desc) as d;

```
+---------------------+--------+
| Most Popular Product | Profit |
+---------------------+--------+
| Color Copy          |    351 |
+---------------------+--------+
1 row in set (0.00 sec)
```

9. Find the repair tech who spends the least amount of time working on equipment.

select min(hours), techName from maintenance;

```
+------------+---------------+
| min(hours) | techName      |
+------------+---------------+
|          1 | Mark Rockroth |
+------------+---------------+
1 row in set (0.00 sec)
```

10. Find busiest day of the year. This is useful for the analysis of our company projects.

select max(q.busiest) as 'jobs on this day', q.year, q.month, q.day from (select count(jnumber) as busiest, extract(year from jdate) as year, extract(month from jdate) as month, extract(day from jdate) as day from job group by year, month, day) as q group by busiest;

```
+------------------+------+-------+------+
| jobs on this day | year | month | day  |
+------------------+------+-------+------+
|                1 | 2012 |     2 |   13 |
|                2 | 2013 |     2 |   21 |
+------------------+------+-------+------+
2 rows in set (0.00 sec)
```

11. Find the most expensive job per year. This is useful for analysis of our company.

select Invoice_Number, Name as cn, extract(year from Date) as year, max(Total) as Most_Expensive_Job from invoices group by year order by year asc;

```
+----------------+-------------+------+--------------------+
| Invoice_Number | cn          | year | Most_Expensive_Job |
+----------------+-------------+------+--------------------+
|          22801 | John Adams  | 2012 |               1.62 |
|          12002 | Econo Foods | 2013 |                324 |
+----------------+-------------+------+--------------------+
2 rows in set (0.00 sec)
```

12. Find the most frequent customer per year. We want to know who we have seen the most during this past year.

```
select max(jn) as "Orders", year, cn as Name from (select custName as cn, extract(year from jdate) as year, count(jNumber) as jn from job group by year, custName order by jn desc) as d group by year order by year desc;
+--------+------+--------------+
| Orders | year | Name         |
+--------+------+--------------+
|      4 | 2013 | Joes Gas N Go |
|      1 | 2012 | Jeff Jepson   |
+--------+------+--------------+
2 rows in set (0.00 sec)
```

Non-Trivial Modifications

UPDATES:

1. Find most popular product and increase the price in the price table. This would be used in an attempt to increase profits.

mysql> update Product_Priced set Price = (Price * 1.10) where Product_Name in (select pn from (select max(pq), pn from(select sum(productQuantity) as pq, productName as pn from jobcontains group by productName order by pq desc) as t) as n);

Before Update:

mysql> select * from Product_Priced;

Product_Name	Paper_Name	Paper_Color	Price
Color Copy	Plain Paper	White	0.49
Spiral Binding	Plain Paper	White	2
Press Printing	Plain Paper	White	0.04
BW Copy	Plain Paper	White	0.08
Folding	Plain Paper	White	0.01
Scoring	Plain Paper	White	0.01
Color Copy	Classic Crest Text	Avalanche White	0.56
Press Printing	Classic Crest Text	Avalanche White	0.11
BW Copy	Classic Crest Text	Avalanche White	0.15
Color Copy	Classic Crest Text	Solar White	0.56
Press Printing	Classic Crest Text	Solar White	0.11
BW Copy	Classic Crest Text	Solar White	0.15
Color Copy	Classic Crest Text	Avon Brilliant White	0.56
Press Printing	Classic Crest Text	Avon Brilliant White	0.11
BW Copy	Classic Crest Text	Avon Brilliant White	0.15
Color Copy	Classic Crest Text	Classic Natural White	0.56
Press Printing	Classic Crest Text	Classic Natural White	0.11
BW Copy	Classic Crest Text	Classic Natural White	0.15
Color Copy	Classic Crest Writing	Antique Gray	0.55
Press Printing	Classic Crest Writing	Antique Gray	0.1
BW Copy	Classic Crest Writing	Antique Gray	0.14
Color Copy	Classic Crest Writing	Avalanche White	0.55
Press Printing	Classic Crest Writing	Avalanche White	0.1
BW Copy	Classic Crest Writing	Avalanche White	0.14
Color Copy	Classic Crest Writing	Avon Brillian White	0.55
Press Printing	Classic Crest Writing	Avon Brillian White	0.1
BW Copy	Classic Crest Writing	Avon Brillian White	0.14
Color Copy	Classic Crest Writing	Baronial Ivory	0.55
Press Printing	Classic Crest Writing	Baronial Ivory	0.1
BW Copy	Classic Crest Writing	Baronial Ivory	0.14
Color Copy	Classic Crest Writing	Classical Cream	0.55
Press Printing	Classic Crest Writing	Classical Cream	0.1
BW Copy	Classic Crest Writing	Classical Cream	0.14
Color Copy	Classic Crest Writing	Classic Natural White	0.55
Press Printing	Classic Crest Writing	Classic Natural White	0.1
BW Copy	Classic Crest Writing	Classic Natural White	0.14
Color Copy	Classic Crest Writing	Earthstone	0.55

Press Printing	Classic Crest Writing	Earthstone	0.1
BW Copy	Classic Crest Writing	Earthstone	0.14
Color Copy	Classic Crest Writing	Millstone	0.55
Press Printing	Classic Crest Writing	Millstone	0.1
BW Copy	Classic Crest Writing	Millstone	0.14
Color Copy	Classic Crest Writing	Potomac Blue	0.55
Press Printing	Classic Crest Writing	Potomac Blue	0.1
BW Copy	Classic Crest Writing	Potomac Blue	0.14
Color Copy	Classic Crest Cover	Avon Brilliant White	0.64
Press Printing	Classic Crest Cover	Avon Brilliant White	0.19
BW Copy	Classic Crest Cover	Avon Brilliant White	0.23
Color Copy	Classic Crest Cover	Classic Natural White	0.64
Press Printing	Classic Crest Cover	Classic Natural White	0.19
BW Copy	Classic Crest Cover	Classic Natural White	0.23
Color Copy	Classic Crest Cover	Solar White	0.64
Press Printing	Classic Crest Cover	Solar White	0.19
BW Copy	Classic Crest Cover	Solar White	0.23
Color Copy	Classic Crest Cover	Antique Gray	0.67
Press Printing	Classic Crest Cover	Antique Gray	0.22
BW Copy	Classic Crest Cover	Antique Gray	0.26
Color Copy	Classic Crest Cover	Avalanche White	0.67
Press Printing	Classic Crest Cover	Avalanche White	0.22
BW Copy	Classic Crest Cover	Avalanche White	0.26
Color Copy	Classic Crest Cover	Avon Brilliant White	0.67
Press Printing	Classic Crest Cover	Avon Brilliant White	0.22
BW Copy	Classic Crest Cover	Avon Brilliant White	0.26
Color Copy	Classic Crest Cover	Baronial Ivory	0.67
Press Printing	Classic Crest Cover	Baronial Ivory	0.22
BW Copy	Classic Crest Cover	Baronial Ivory	0.26
Color Copy	Classic Crest Cover	Canyon Brown	0.73
Press Printing	Classic Crest Cover	Canyon Brown	0.28
BW Copy	Classic Crest Cover	Canyon Brown	0.32
Color Copy	Classic Crest Cover	Classic Cream	0.67
Press Printing	Classic Crest Cover	Classic Cream	0.22
BW Copy	Classic Crest Cover	Classic Cream	0.26
Color Copy	Classic Crest Cover	Classic Natural White	0.67
Press Printing	Classic Crest Cover	Classic Natural White	0.22
BW Copy	Classic Crest Cover	Classic Natural White	0.26
Color Copy	Classic Crest Cover	Earthstone	0.68
Press Printing	Classic Crest Cover	Earthstone	0.23
BW Copy	Classic Crest Cover	Earthstone	0.27
Color Copy	Classic Crest Cover	Epic Black	0.73
Press Printing	Classic Crest Cover	Epic Black	0.28
BW Copy	Classic Crest Cover	Epic Black	0.32

81 rows in set (0.00 sec)

After update:

mysql> select * from Product_Priced;

Product_Name	Paper_Name	Paper_Color	Price
Color Copy	Plain Paper	White	0.539
Spiral Binding	Plain Paper	White	2
Press Printing	Plain Paper	White	0.04
BW Copy	Plain Paper	White	0.08
Folding	Plain Paper	White	0.01
Scoring	Plain Paper	White	0.01
Color Copy	Classic Crest Text	Avalanche White	0.616
Press Printing	Classic Crest Text	Avalanche White	0.11
BW Copy	Classic Crest Text	Avalanche White	0.15
Color Copy	Classic Crest Text	Solar White	0.616
Press Printing	Classic Crest Text	Solar White	0.11
BW Copy	Classic Crest Text	Solar White	0.15
Color Copy	Classic Crest Text	Avon Brilliant White	0.616
Press Printing	Classic Crest Text	Avon Brilliant White	0.11
BW Copy	Classic Crest Text	Avon Brilliant White	0.15
Color Copy	Classic Crest Text	Classic Natural White	0.616
Press Printing	Classic Crest Text	Classic Natural White	0.11
BW Copy	Classic Crest Text	Classic Natural White	0.15
Color Copy	Classic Crest Writing	Antique Gray	0.605
Press Printing	Classic Crest Writing	Antique Gray	0.1
BW Copy	Classic Crest Writing	Antique Gray	0.14
Color Copy	Classic Crest Writing	Avalanche White	0.605
Press Printing	Classic Crest Writing	Avalanche White	0.1
BW Copy	Classic Crest Writing	Avalanche White	0.14
Color Copy	Classic Crest Writing	Avon Brillian White	0.605
Press Printing	Classic Crest Writing	Avon Brillian White	0.1
BW Copy	Classic Crest Writing	Avon Brillian White	0.14
Color Copy	Classic Crest Writing	Baronial Ivory	0.605
Press Printing	Classic Crest Writing	Baronial Ivory	0.1
BW Copy	Classic Crest Writing	Baronial Ivory	0.14
Color Copy	Classic Crest Writing	Classical Cream	0.605
Press Printing	Classic Crest Writing	Classical Cream	0.1
BW Copy	Classic Crest Writing	Classical Cream	0.14
Color Copy	Classic Crest Writing	Classic Natural White	0.605
Press Printing	Classic Crest Writing	Classic Natural White	0.1
BW Copy	Classic Crest Writing	Classic Natural White	0.14
Color Copy	Classic Crest Writing	Earthstone	0.605
Press Printing	Classic Crest Writing	Earthstone	0.1
BW Copy	Classic Crest Writing	Earthstone	0.14
Color Copy	Classic Crest Writing	Millstone	0.605
Press Printing	Classic Crest Writing	Millstone	0.1
BW Copy	Classic Crest Writing	Millstone	0.14
Color Copy	Classic Crest Writing	Potomac Blue	0.605
Press Printing	Classic Crest Writing	Potomac Blue	0.1
BW Copy	Classic Crest Writing	Potomac Blue	0.14
Color Copy	Classic Crest Cover	Avon Brilliant White	0.704

Press Printing	Classic Crest Cover	Avon Brilliant White	0.19
BW Copy	Classic Crest Cover	Avon Brilliant White	0.23
Color Copy	Classic Crest Cover	Classic Natural White	0.704
Press Printing	Classic Crest Cover	Classic Natural White	0.19
BW Copy	Classic Crest Cover	Classic Natural White	0.23
Color Copy	Classic Crest Cover	Solar White	0.704
Press Printing	Classic Crest Cover	Solar White	0.19
BW Copy	Classic Crest Cover	Solar White	0.23
Color Copy	Classic Crest Cover	Antique Gray	0.737
Press Printing	Classic Crest Cover	Antique Gray	0.22
BW Copy	Classic Crest Cover	Antique Gray	0.26
Color Copy	Classic Crest Cover	Avalanche White	0.737
Press Printing	Classic Crest Cover	Avalanche White	0.22
BW Copy	Classic Crest Cover	Avalanche White	0.26
Color Copy	Classic Crest Cover	Avon Brilliant White	0.737
Press Printing	Classic Crest Cover	Avon Brilliant White	0.22
BW Copy	Classic Crest Cover	Avon Brilliant White	0.26
Color Copy	Classic Crest Cover	Baronial Ivory	0.737
Press Printing	Classic Crest Cover	Baronial Ivory	0.22
BW Copy	Classic Crest Cover	Baronial Ivory	0.26
Color Copy	Classic Crest Cover	Canyon Brown	0.803
Press Printing	Classic Crest Cover	Canyon Brown	0.28
BW Copy	Classic Crest Cover	Canyon Brown	0.32
Color Copy	Classic Crest Cover	Classic Cream	0.737
Press Printing	Classic Crest Cover	Classic Cream	0.22
BW Copy	Classic Crest Cover	Classic Cream	0.26
Color Copy	Classic Crest Cover	Classic Natural White	0.737
Press Printing	Classic Crest Cover	Classic Natural White	0.22
BW Copy	Classic Crest Cover	Classic Natural White	0.26
Color Copy	Classic Crest Cover	Earthstone	0.748
Press Printing	Classic Crest Cover	Earthstone	0.23
BW Copy	Classic Crest Cover	Earthstone	0.27
Color Copy	Classic Crest Cover	Epic Black	0.803
Press Printing	Classic Crest Cover	Epic Black	0.28
BW Copy	Classic Crest Cover	Epic Black	0.32

81 rows in set (0.01 sec)

2. Find Least Productive employees and reduce salary. This Update selects the lowest performing employees based on output of work and reduces the salary by a percentage. This update likely wouldn't be applied like this in real life unless there was a commission type position and the salary was based on a percentage of the sales in a month.

```
mysql> update position set sSalary = sSalary - (sSalary * .03) where pNumber in(select pNum from(select pNum, count(ssn) from employee, workson, position where ssn = essn and pNum = pNumber group by ssn having count(ssn) < 3)x);
```

Before update:

```
mysql> select * from position;
```

```
+---------+-------------+------------+---------+
| pNumber | pName     | startDate  | sSalary |
+---------+-------------+------------+---------+
|  123456 | Printer       | 2013-02-28 |   30000 |
|  234567 | Typesetter  | 2012-03-13 |   33000 |
|  345678 | Manager   | 1999-05-15 |   43000 |
|  456789 | Copy Tech   | 2012-08-11 |   25000 |
|  567890 | Copy Tech 2 | 2013-01-20 |   30000 |
|  678901 | Trainee     | 2013-02-27 |   20000 |
|  789012 | Trainee     | 2013-03-01 |   20000 |
|  890123 | Copy Tech   | 2012-09-22 |   25000 |
+---------+-------------+------------+---------+
8 rows in set (0.00 sec)
```

After update:

```
mysql> select * from position;
+---------+-------------+------------+---------+
| pNumber | pName     | startDate  | sSalary |
+---------+-------------+------------+---------+
|  123456 | Printer       | 2013-02-28 |   29100 |
|  234567 | Typesetter  | 2012-03-13 |   32010 |
|  345678 | Manager   | 1999-05-15 |   43000 |
|  456789 | Copy Tech   | 2012-08-11 |   25000 |
|  567890 | Copy Tech 2 | 2013-01-20 |   29100 |
|  678901 | Trainee     | 2013-02-27 |   20000 |
|  789012 | Trainee     | 2013-03-01 |   20000 |
|  890123 | Copy Tech   | 2012-09-22 |   25000 |
+---------+-------------+------------+---------+
8 rows in set (0.00 sec)
```

3. Find the most productive employees and increase their salary a percentage based on performance measured by number of jobs performed. This update could be applied to a bonus program in real life.

```
mysql> update position set sSalary = sSalary + (sSalary * .03) where pNumber in(select pNum from(select pNum, count(ssn) from employee, workson, position where ssn = essn and pNum = pNumber group by ssn having count(ssn) > 3)x);
```

Before update:

```
mysql> select * from position;
+---------+-------------+------------+---------+
| pNumber | pName     | startDate  | sSalary |
+---------+-------------+------------+---------+
|  123456 | Printer       | 2013-02-28 |   29100 |
|  234567 | Typesetter  | 2012-03-13 |   32010 |
|  345678 | Manager   | 1999-05-15 |   43000 |
|  456789 | Copy Tech   | 2012-08-11 |   25000 |
|  567890 | Copy Tech 2 | 2013-01-20 |   29100 |
|  678901 | Trainee     | 2013-02-27 |   20000 |
|  789012 | Trainee     | 2013-03-01 |   20000 |
```

```
|  890123 | Copy Tech   | 2012-09-22 |   25000 |
+---------+-------------+------------+---------+
8 rows in set (0.00 sec)
```

After update:

```
mysql> select * from position;
+---------+-------------+------------+---------+
| pNumber | pName     | startDate  | sSalary |
+---------+-------------+------------+---------+
|  123456 | Printer       | 2013-02-28 |   29100 |
|  234567 | Typesetter  | 2012-03-13 |   32010 |
|  345678 | Manager   | 1999-05-15 |   44290 |
|  456789 | Copy Tech   | 2012-08-11 |   25750 |
|  567890 | Copy Tech 2 | 2013-01-20 |   29100 |
|  678901 | Trainee     | 2013-02-27 |   20000 |
|  789012 | Trainee     | 2013-03-01 |   20000 |
|  890123 | Copy Tech   | 2012-09-22 |   25000 |
+---------+-------------+------------+---------+
8 rows in set (0.00 sec)
```

4. Find all invoices associated with a single customer and change status from unpaid to paid. Sometimes a customer will wish to pay off all their outstanding invoices at one time. This would make it easy to take care of that in one transaction.

```
mysql> update invoices set Paid = 'TRUE' where Name = 'Joes Gas N Go';
```

Before Update:

```
+----------------+------------+------------+--------------+-------------------------------------+----------+-------+--------+-------+
| Invoice_Number | Date       | Remit_by   | Name         | cAddress                            | Subtotal | Tax   | Total  | Paid  |
+----------------+------------+------------+--------------+-------------------------------------+----------+-------+--------+-------+
|          12502 | 2013-01-02 | 2013-02-14 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |       42 |  3.36 |  45.36 | FALSE |
|          13103 | 2013-01-31 | 2013-03-07 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |       55 |   4.4 |   59.4 | FALSE |
|          21801 | 2013-02-18 | 2013-05-01 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |       28 |  2.24 |  30.24 | TRUE  |
|          22102 | 2013-02-21 | 2013-05-02 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |     3205 | 256.4 | 3461.4 | FALSE |
+----------------+------------+------------+--------------+-------------------------------------+----------+-------+--------+-------+
4 rows in set (0.00 sec)
```

After update:

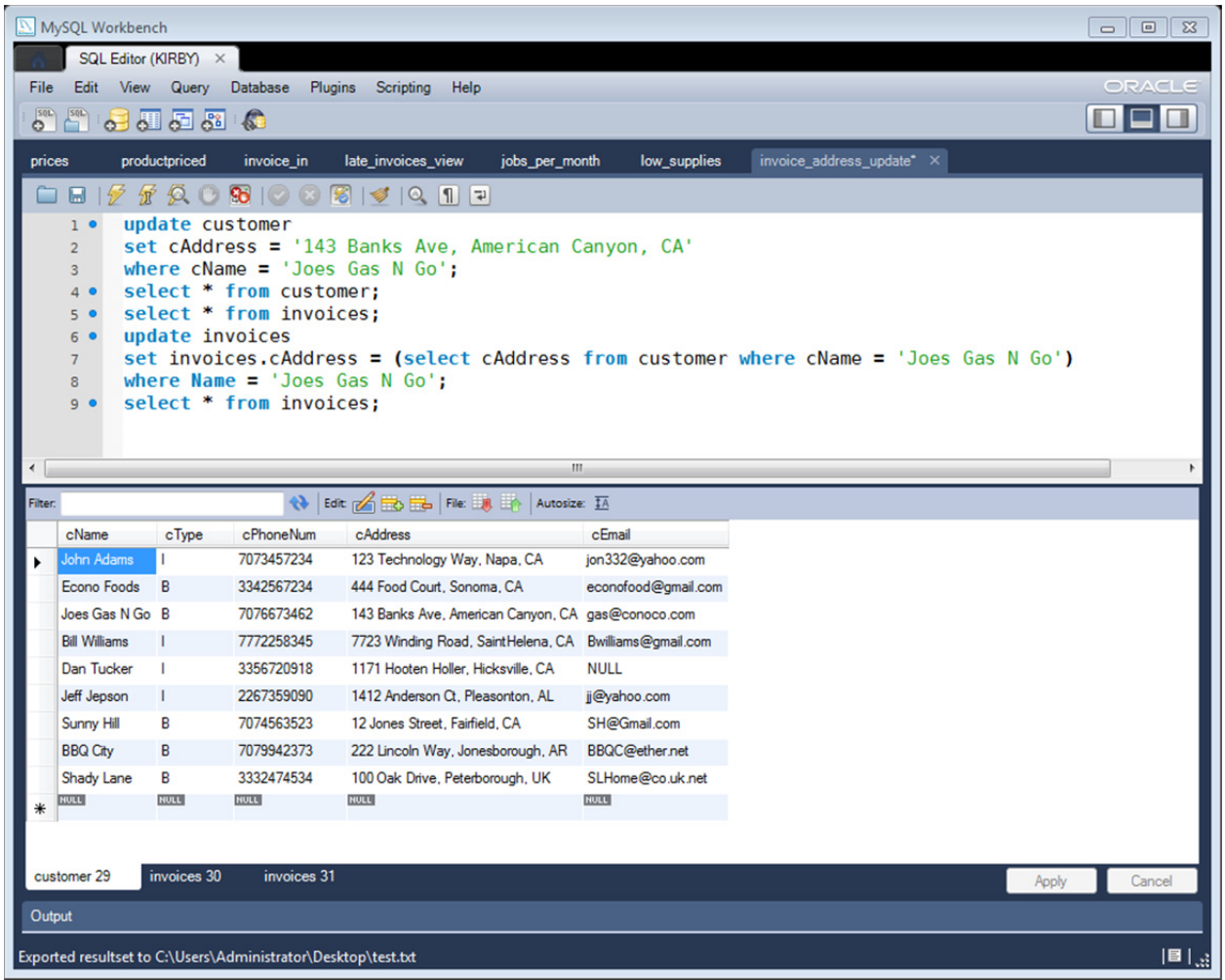
```
+----------------+------------+------------+--------------+-------------------------------------+----------+-------+--------+------+
| Invoice_Number | Date       | Remit_by   | Name         | cAddress                            | Subtotal | Tax   | Total  | Paid |
+----------------+------------+------------+--------------+-------------------------------------+----------+-------+--------+------+
|          12502 | 2013-01-02 | 2013-02-14 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |       42 |  3.36 |  45.36 | TRUE |
|          13103 | 2013-01-31 | 2013-03-07 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |       55 |   4.4 |   59.4 | TRUE |
|          21801 | 2013-02-18 | 2013-05-01 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |       28 |  2.24 |  30.24 | TRUE |
|          22102 | 2013-02-21 | 2013-05-02 | Joes Gas N Go | 73 Fuel Stop Ave, American Canyon, CA |     3205 | 256.4 | 3461.4 | TRUE |
+----------------+------------+------------+--------------+-------------------------------------+----------+-------+--------+------+
4 rows in set (0.00 sec)
```

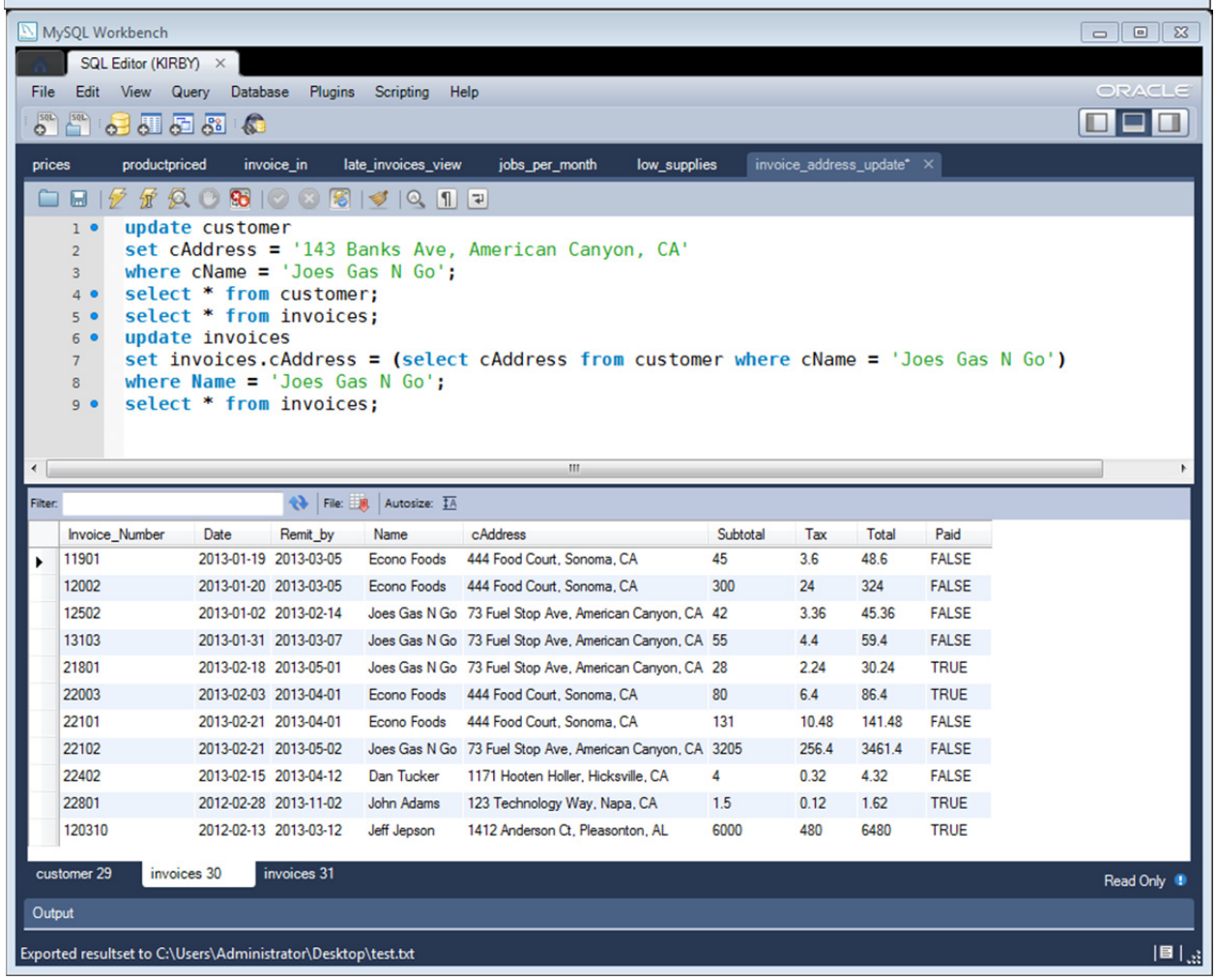
5. Find all invoices associated with a particular customer who has changed their address and update the address on the invoices with the address queried from the customer table.

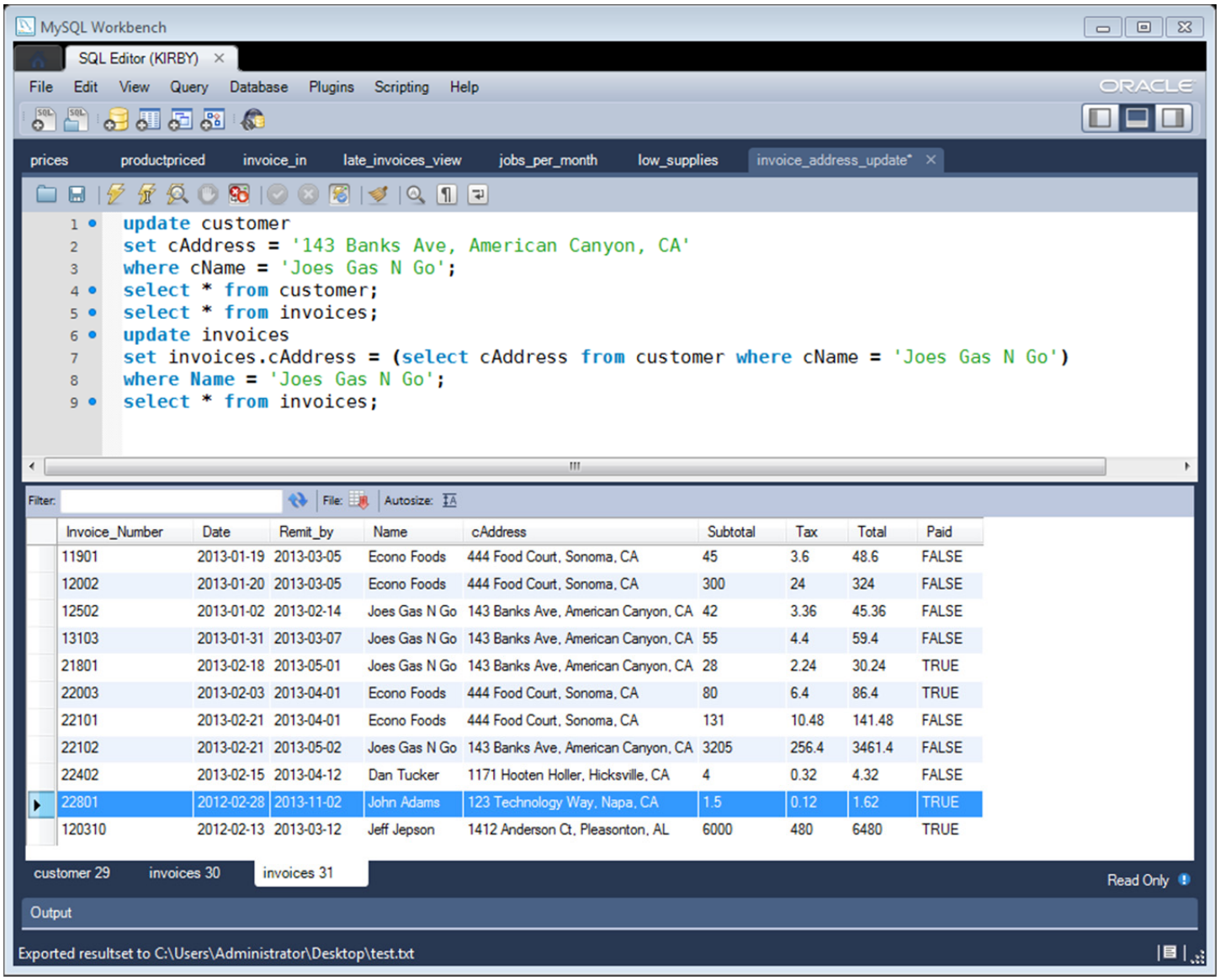
--Creating the address change in the customer table

```
update customer
set cAddress = '143 Banks Ave, American Canyon, CA'
```

```
where cName = 'Joes Gas N Go';

--Comparing the address mismatch
select * from customer;
select * from invoices;

--Updating all invoices for that customer with the new address queried from the customer table
update invoices
set invoices.cAddress = (select cAddress from customer where cName = 'Joes Gas N Go')
where Name = 'Joes Gas N Go';

--Confirming it worked
select * from invoices;
```

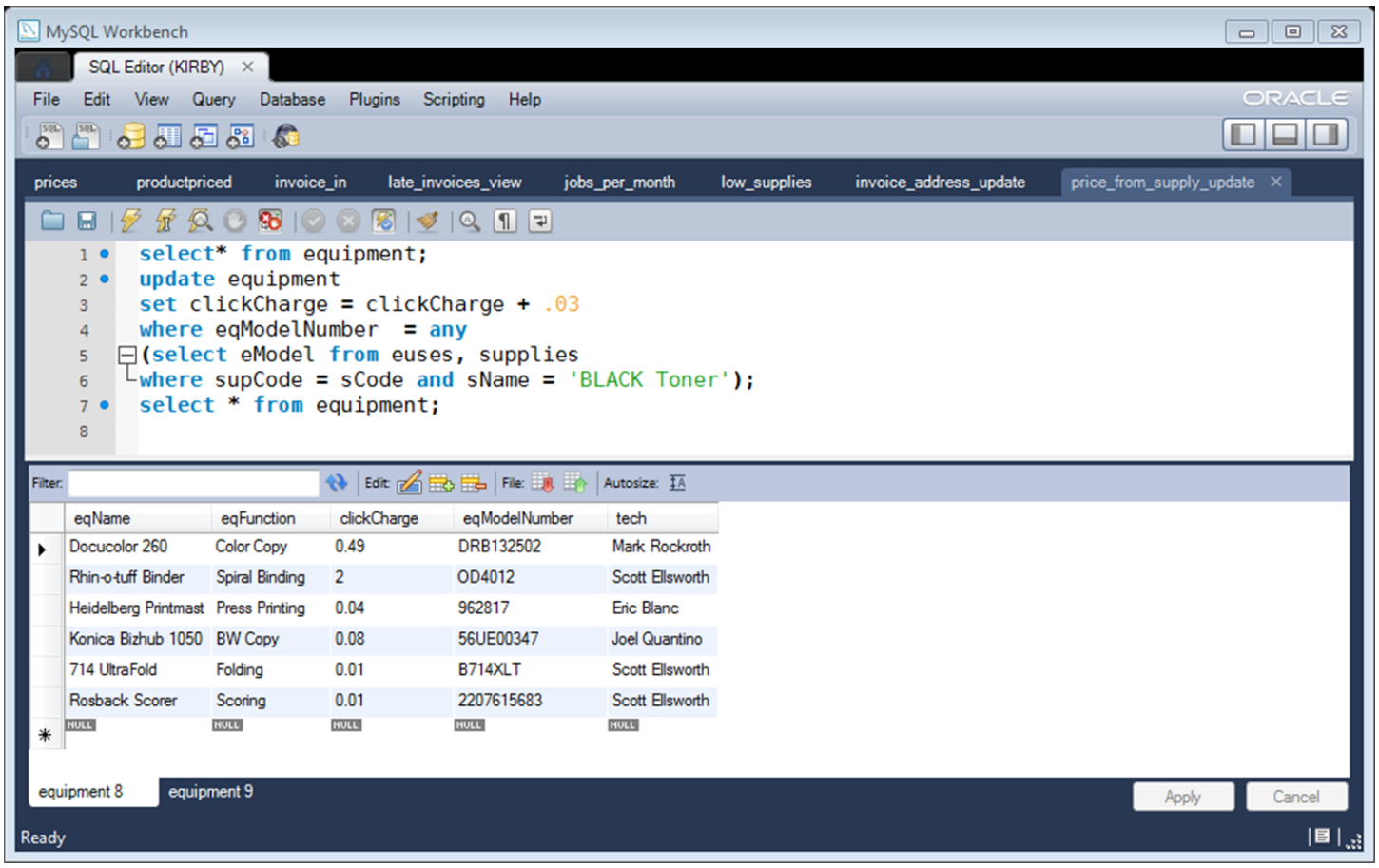

MySQL Workbench
SQL Editor (KIRBY)
File Edit View Query Database Plugins Scripting Help
ORACLE
prices productpriced invoice_in late_invoices_view jobs_per_month low_supplies invoice_address_update*
1 update customer
2 set cAddress = '143 Banks Ave, American Canyon, CA'
3 where cName = 'Joes Gas N Go';
4 select * from customer;
5 select * from invoices;
6 update invoices
7 set invoices.cAddress = (select cAddress from customer where cName = 'Joes Gas N Go')
8 where Name = 'Joes Gas N Go';
9 select * from invoices;
Filter: Edit: File: Autosize:
cName cType cPhoneNum cAddress cEmail
John Adams I 7073457234 123 Technology Way, Napa, CA jon332@yahoo.com
Econo Foods B 3342567234 444 Food Court, Sonoma, CA econofood@gmail.com
Joes Gas N Go B 7076673462 143 Banks Ave, American Canyon, CA gas@conoco.com
Bill Williams I 7772258345 7723 Winding Road, SaintHelena, CA Bwilliams@gmail.com
Dan Tucker I 3356720918 1171 Hooten Holler, Hicksville, CA NULL
Jeff Jepson I 2267359090 1412 Anderson Ct, Pleasonton, AL jj@yahoo.com
Sunny Hill B 7074563523 12 Jones Street, Fairfield, CA SH@Gmail.com
BBQ City B 7079942373 222 Lincoln Way, Jonesborough, AR BBQC@ether.net
Shady Lane B 3332474534 100 Oak Drive, Peterborough, UK SLHome@co.uk.net
NULL NULL NULL NULL NULL
customer 29 invoices 30 invoices 31
Apply Cancel
Output
Exported resultset to C:\Users\Administrator\Desktop\test.txt

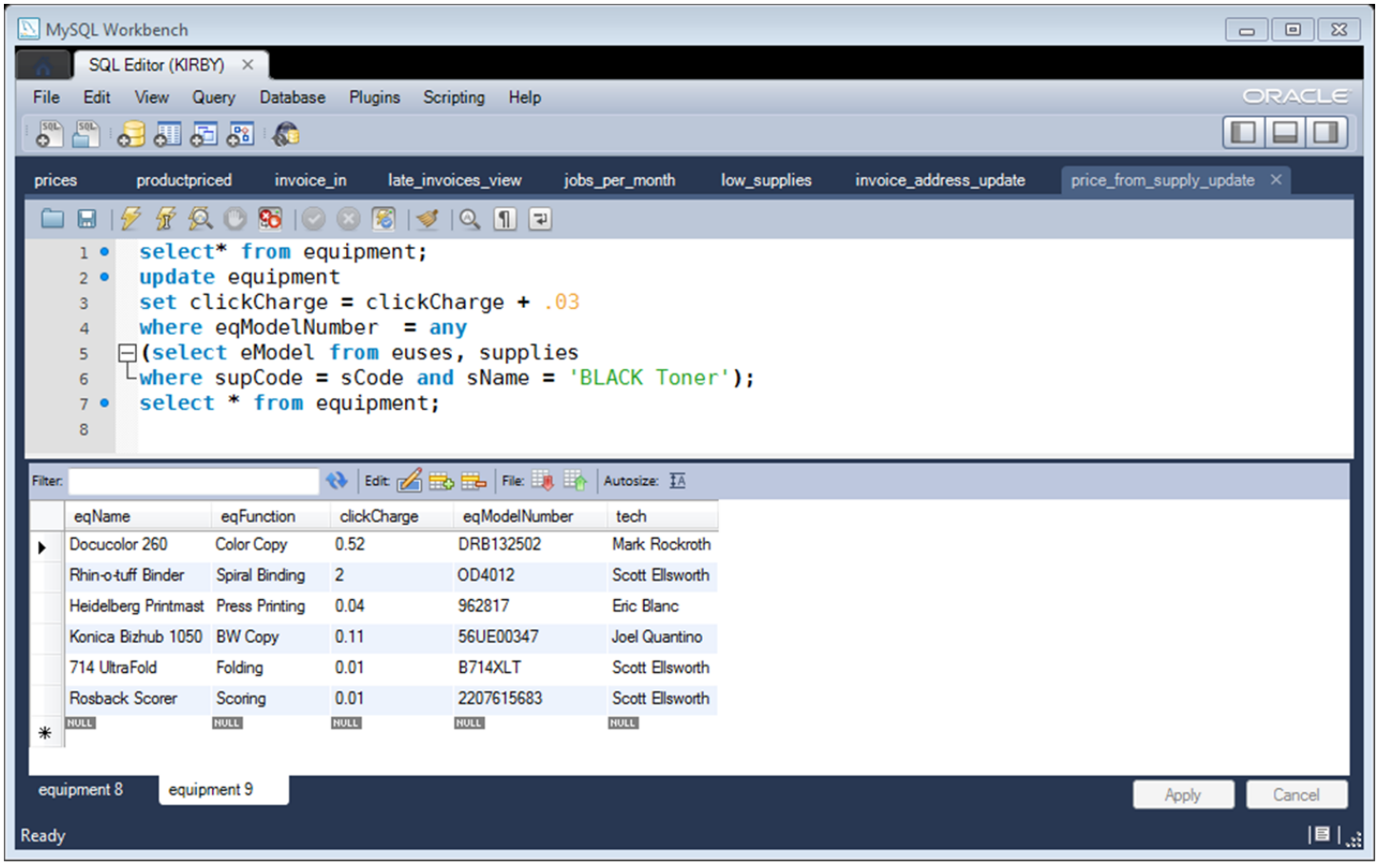

MySQL Workbench
SQL Editor (KIRBY)
File Edit View Query Database Plugins Scripting Help
ORACLE
prices productpriced invoice_in late_invoices_view jobs_per_month low_supplies invoice_address_update*
1 update customer
2 set cAddress = '143 Banks Ave, American Canyon, CA'
3 where cName = 'Joes Gas N Go';
4 select * from customer;
5 select * from invoices;
6 update invoices
7 set invoices.cAddress = (select cAddress from customer where cName = 'Joes Gas N Go')
8 where Name = 'Joes Gas N Go';
9 select * from invoices;
Filter: File: Autosize:
Invoice_Number Date Remit_by Name cAddress Subtotal Tax Total Paid
11901 2013-01-19 2013-03-05 Econo Foods 444 Food Court, Sonoma, CA 45 3.6 48.6 FALSE
12002 2013-01-20 2013-03-05 Econo Foods 444 Food Court, Sonoma, CA 300 24 324 FALSE
12502 2013-01-02 2013-02-14 Joes Gas N Go 73 Fuel Stop Ave, American Canyon, CA 42 3.36 45.36 FALSE
13103 2013-01-31 2013-03-07 Joes Gas N Go 73 Fuel Stop Ave, American Canyon, CA 55 4.4 59.4 FALSE
21801 2013-02-18 2013-05-01 Joes Gas N Go 73 Fuel Stop Ave, American Canyon, CA 28 2.24 30.24 TRUE
22003 2013-02-03 2013-04-01 Econo Foods 444 Food Court, Sonoma, CA 80 6.4 86.4 TRUE
22101 2013-02-21 2013-04-01 Econo Foods 444 Food Court, Sonoma, CA 131 10.48 141.48 FALSE
22102 2013-02-21 2013-05-02 Joes Gas N Go 73 Fuel Stop Ave, American Canyon, CA 3205 256.4 3461.4 FALSE
22402 2013-02-15 2013-04-12 Dan Tucker 1171 Hooten Holler, Hicksville, CA 4 0.32 4.32 FALSE
22801 2012-02-28 2013-11-02 John Adams 123 Technology Way, Napa, CA 1.5 0.12 1.62 TRUE
120310 2012-02-13 2013-03-12 Jeff Jepson 1412 Anderson Ct, Pleasonton, AL 6000 480 6480 TRUE
customer 29 invoices 30 invoices 31
Read Only
Output
Exported resultset to C:\Users\Administrator\Desktop\test.txt



6. Find all equipment that uses a particular supply and increase the click charges of the equipment by 3 cents. In this instance, the supply used is BLACK Toner.

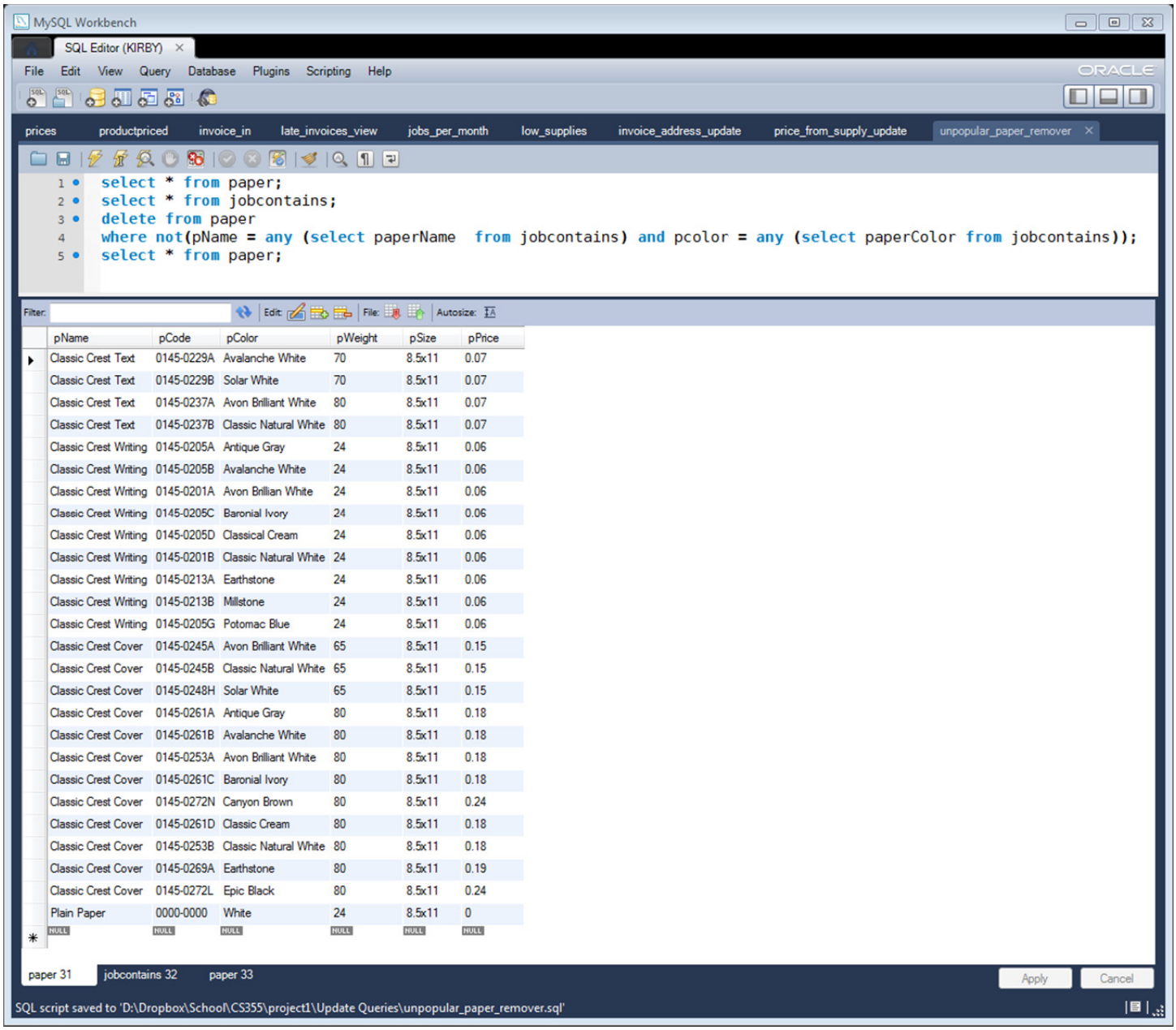
```
select* from equipment;
update equipment
set clickCharge = clickCharge + .03
where eqModelNumber  = any
(select eModel from euses, supplies
where supCode = sCode and sName = 'BLACK Toner');
select * from equipment;
```

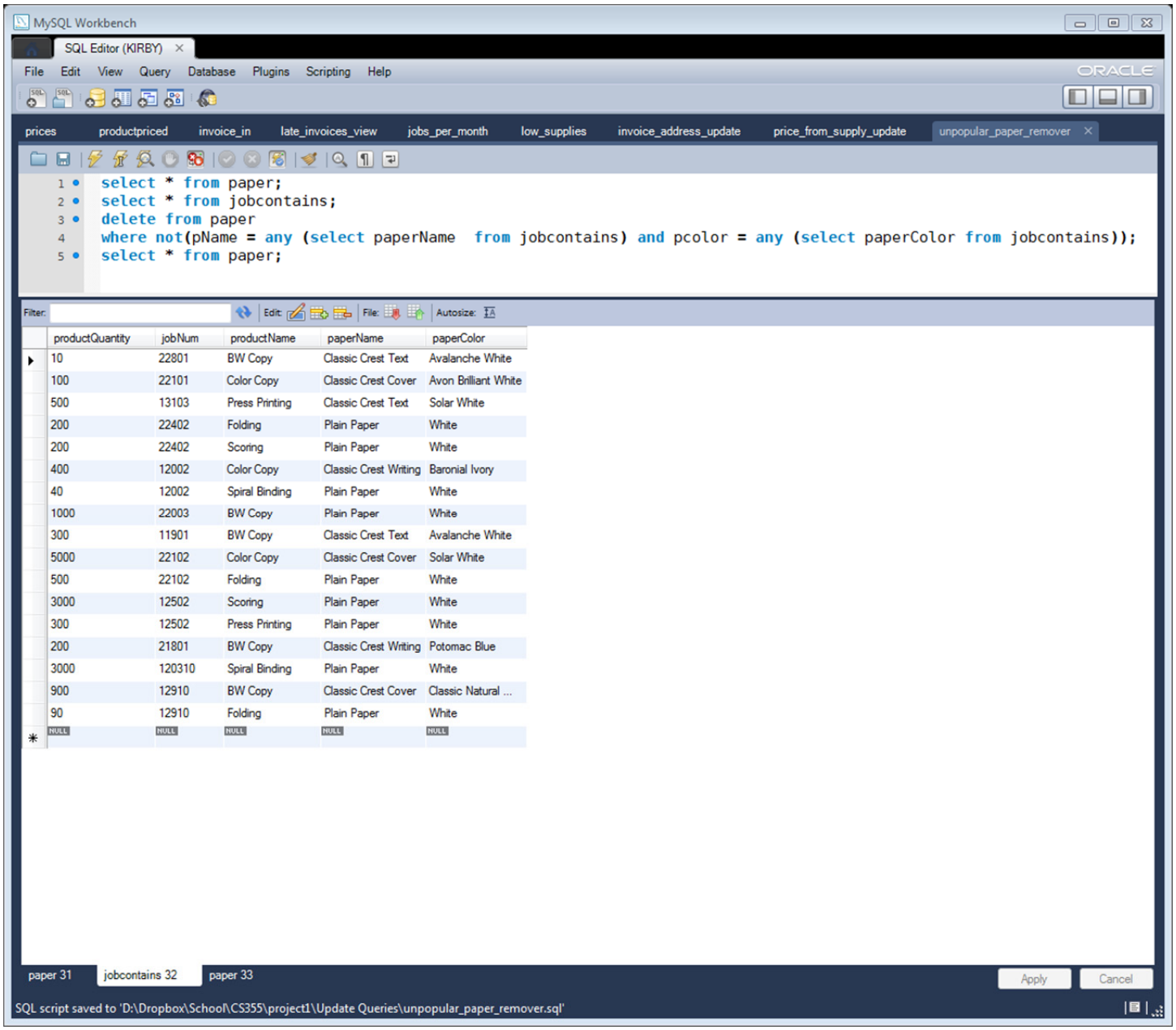

MySQL Workbench
SQL Editor (KIRBY)
File Edit View Query Database Plugins Scripting Help
ORACLE
prices productpriced invoice_in late_invoices_view jobs_per_month low_supplies invoice_address_update price_from_supply_update
1 select* from equipment;
2 update equipment
3 set clickCharge = clickCharge + .03
4 where eqModelNumber = any
5 (select eModel from euses, supplies
6 where supCode = sCode and sName = 'BLACK Toner');
7 select * from equipment;
8
Filter: Edit: File: Autosize:
eqName eqFunction clickCharge eqModelNumber tech
Docucolor 260 Color Copy 0.49 DRB132502 Mark Rockroth
Rhin-o-tuff Binder Spiral Binding 2 OD4012 Scott Ellsworth
Heidelberg Printmast Press Printing 0.04 962817 Eric Blanc
Konica Bizhub 1050 BW Copy 0.08 56UE00347 Joel Quantino
714 UltraFold Folding 0.01 B714XLT Scott Ellsworth
Rosback Scorer Scoring 0.01 2207615683 Scott Ellsworth
NULL NULL NULL NULL NULL
equipment 8 equipment 9
Apply Cancel
Ready

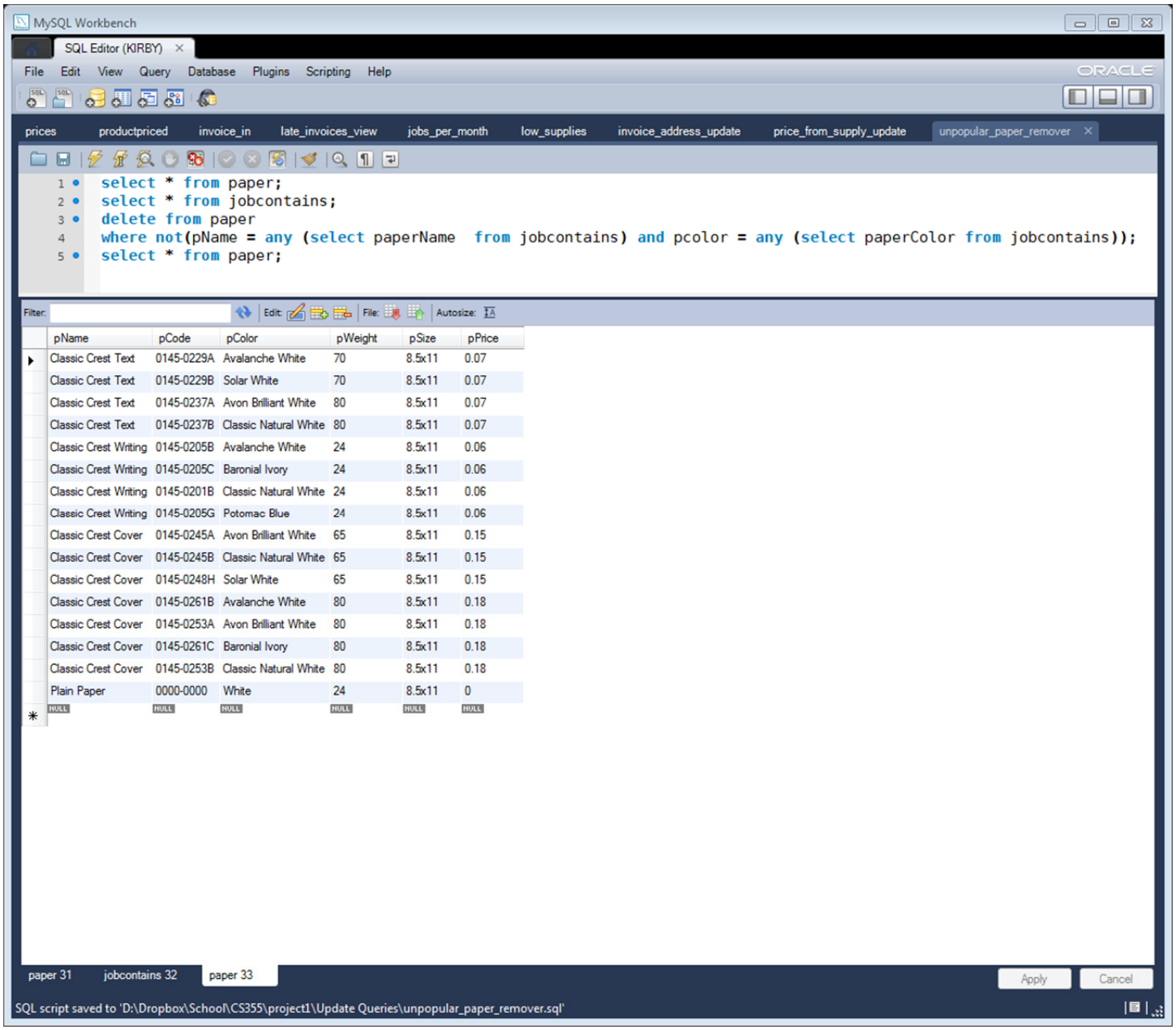

MySQL Workbench
SQL Editor (KIRBY)
File Edit View Query Database Plugins Scripting Help
ORACLE
prices productpriced invoice_in late_invoices_view jobs_per_month low_supplies invoice_address_update price_from_supply_update
1 select* from equipment;
2 update equipment
3 set clickCharge = clickCharge + .03
4 where eqModelNumber = any
5 (select eModel from euses, supplies
6 where supCode = sCode and sName = 'BLACK Toner');
7 select * from equipment;
8
Filter: Edit: File: Autosize:
eqName eqFunction clickCharge eqModelNumber tech
Docucolor 260 Color Copy 0.52 DRB132502 Mark Rockroth
Rhin-o-tuff Binder Spiral Binding 2 OD4012 Scott Ellsworth
Heidelberg Printmast Press Printing 0.04 962817 Eric Blanc
Konica Bizhub 1050 BW Copy 0.11 56UE00347 Joel Quantino
714 UltraFold Folding 0.01 B714XLT Scott Ellsworth
Rosback Scorer Scoring 0.01 2207615683 Scott Ellsworth
NULL NULL NULL NULL NULL
equipment 8 equipment 9
Apply Cancel
Ready

7. Delete all Paper Name + Color combinations that nobody has bought before. With all the different kinds of paper out there, the stuff people don't like piles up quickly. This will remove unwanted paper from the database.

```
select * from paper;
select * from jobcontains;
delete from paper
where not(pName = any (select paperName  from jobcontains) and pcolor = any (select paperColor from jobcontains));
select * from paper;
```




MySQL Workbench
SQL Editor (KIRBY)
File Edit View Query Database Plugins Scripting Help
ORACLE
prices productpriced invoice_in late_invoices_view jobs_per_month low_supplies invoice_address_update price_from_supply_update unpopular_paper_remover
1 select * from paper;
2 select * from jobcontains;
3 delete from paper
4 where not(pName = any (select paperName from jobcontains) and pcolor = any (select paperColor from jobcontains));
5 select * from paper;
Filter:
Edit:
File:
Autosize:
productQuantity jobNum productName paperName paperColor
10 22801 BW Copy Classic Crest Text Avalanche White
100 22101 Color Copy Classic Crest Cover Avon Brilliant White
500 13103 Press Printing Classic Crest Text Solar White
200 22402 Folding Plain Paper White
200 22402 Scoring Plain Paper White
400 12002 Color Copy Classic Crest Writing Baronial Ivory
40 12002 Spiral Binding Plain Paper White
1000 22003 BW Copy Plain Paper White
300 11901 BW Copy Classic Crest Text Avalanche White
5000 22102 Color Copy Classic Crest Cover Solar White
500 22102 Folding Plain Paper White
3000 12502 Scoring Plain Paper White
300 12502 Press Printing Plain Paper White
200 21801 BW Copy Classic Crest Writing Potomac Blue
3000 120310 Spiral Binding Plain Paper White
900 12910 BW Copy Classic Crest Cover Classic Natural ...
90 12910 Folding Plain Paper White
NULL NULL NULL NULL NULL
paper 31 jobcontains 32 paper 33
Apply Cancel
SQL script saved to 'D:\Dropbox\School\CS355\project1\Update Queries\unpopular_paper_remover.sql'



8. This update checks the status of the supplies when a person is ordering (possibly run by a trigger in the future), checks the onhand amount verses the ordered amount and it compares that to the maximum values allowed and adjusts the order amount accordingly.

update orderedfrom set oQuantity=@excess where spCode in (select spCode from (select spCode, @excess := (sQuantity + oQuantity - sQuantityMax), oQuantity, sQuantity, sQuantityMax from supplies, orderedfrom where (oQuantity + sQuantity) > sQuantityMax and sCode = spCode group by sCode) as o);
Query OK, 1 row affected (0.00 sec)
Rows matched: 1 Changed: 1 Warnings: 0

BEFORE UPDATE:

mysql> select * from orderedfrom;
+--------+----------------+------------+-----------+--------+
| spCode | spName | oDate | oQuantity | sPrice |

```
+--------+----------------+------------+-----------+--------+
| BTO    | Deep Toner     | 0000-00-00 |      2000 |   0.01 |
| MTO    | Deep Toner     | 0000-00-00 |      5000 |   0.05 |
| CTO    | Deep Toner     | 0000-00-00 |      1000 |   0.05 |
| YTO    | Officemax      | 0000-00-00 |      1500 |   0.05 |
| SPS    | Bindings are us | 0000-00-00 |       100 |    0.5 |
+--------+----------------+------------+-----------+--------+
5 rows in set (0.00 sec)
```

AFTER UPDATE

```
mysql> select * from orderedfrom;
+--------+----------------+------------+-----------+--------+
| spCode | spName         | oDate      | oQuantity | sPrice |
+--------+----------------+------------+-----------+--------+
| BTO    | Deep Toner     | 0000-00-00 |      2000 |   0.01 |
| MTO    | Deep Toner     | 0000-00-00 |      4500 |   0.05 |
| CTO    | Deep Toner     | 0000-00-00 |      1000 |   0.05 |
| YTO    | Officemax      | 0000-00-00 |      1500 |   0.05 |
| SPS    | Bindings are us | 0000-00-00 |       100 |    0.5 |
+--------+----------------+------------+-----------+--------+
5 rows in set (0.00 sec)
```

9. This Alter table creates a new column in the position table to add a bonus plan to the company. This one was created to demonstrate the next update which will use the bonus column.

```
mysql> alter table position add bonus INT;
```

10. This Update selects employees who worked on jobs totaling more than $3000. Then calculates 1% of the actual total and gives them a bonus of 1% of the total sale. This could be an annual update for a bonus plan for the company.

```
mysql> update position set bonus = (select case when sum(Total) > 3000 then (sum(Total) * 0.01) else 0 end
from invoices, workson, employee where Invoice_Number=jNum and essn=ssn and pNum=pNumber );
```

Non-Trivial Views

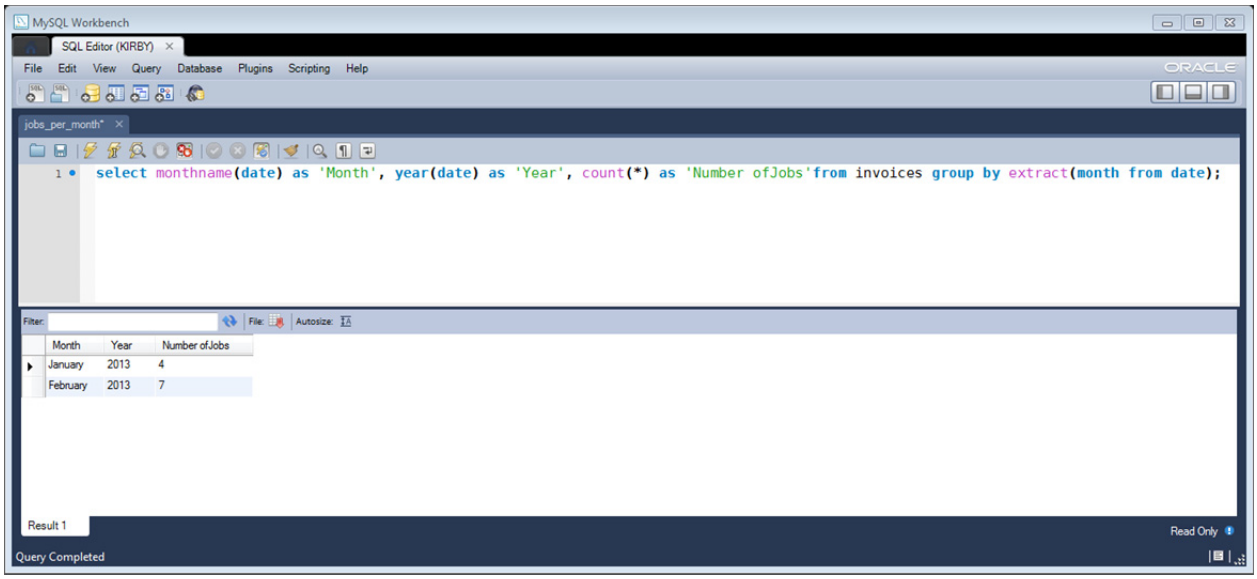
Views

View 1.

The purpose of this view is to display the number of jobs each month. This way we can determine which months are the busiest and adjust inventory and staffing hours accordingly.

Create View jobs_per_month as

select monthname(date) as 'Month', year(date) as 'Year', count(*) as 'Number ofJobs'from invoices group by extract(month from date);



View 2.

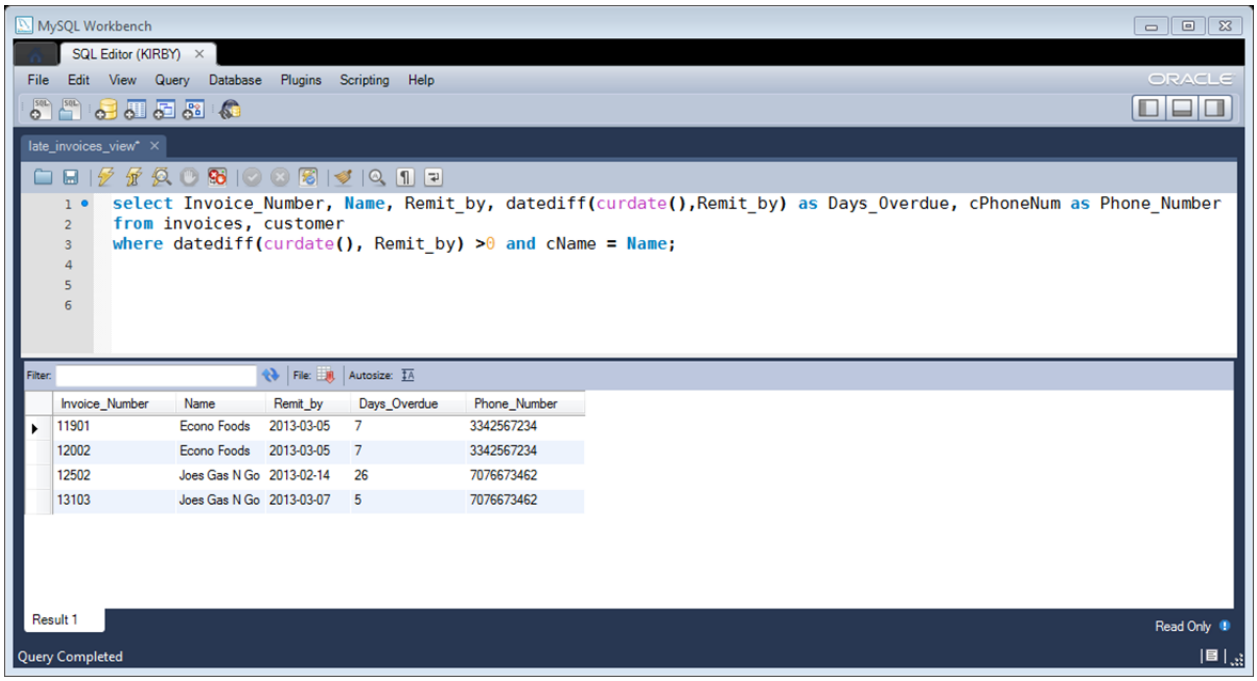
The purpose of this view is to display invoices that are past the remittance date. Often times a quick phone call can get a customer to pay their past due invoice.

CREATE VIEW late_invoices as

select Invoice_Number, Name, Remit_by, datediff(curdate(),Remit_by) as Days_Overdue, cPhoneNum as Phone_Number

from invoices, customer

where datediff(curdate(), Remit_by) >0 and cName = Name;



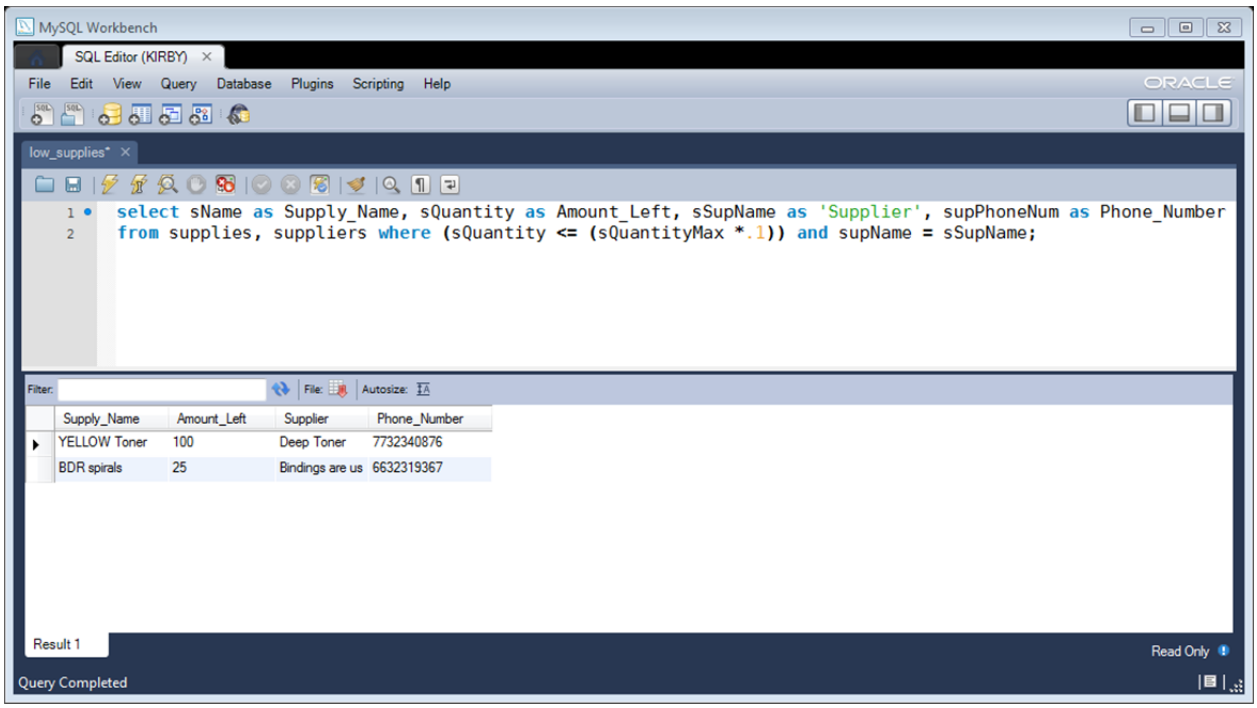
View 3.

This view displays the supplies that are running low and the suppliers to purchase them from. This is useful because if we completely run out of a certain supply, the machine that needs it is useless until we resupply it.

create view low_supplies as

select sName as Supply_Name, sQuantity as Amount_Left, sSupName as 'Supplier', supPhoneNum as Phone_Number

from supplies, suppliers where (sQuantity <= (sQuantityMax *.1)) and supName = sSupName;



Indices

Indices

These indices were selected based on their complexity and/or their frequency of use.

```
CREATE INDEX tech_name_idx ON mainttech (tName);

CREATE INDEX emp_ssn_position_idx ON employee (ssn, pNum);

CREATE INDEX customer_name_idx ON customer (cName);

CREATE INDEX paper_name_paper_color_idx ON product_priced (Paper_Name, Paper_Color);

CREATE INDEX invoice_numbers_idx ON invoices (Invoice_Number);

CREATE INDEX paper_codes_idx ON paper (pCode);
```