IT450 Final Project

Designing, Building, Populating, and Testing a Catering Company Database

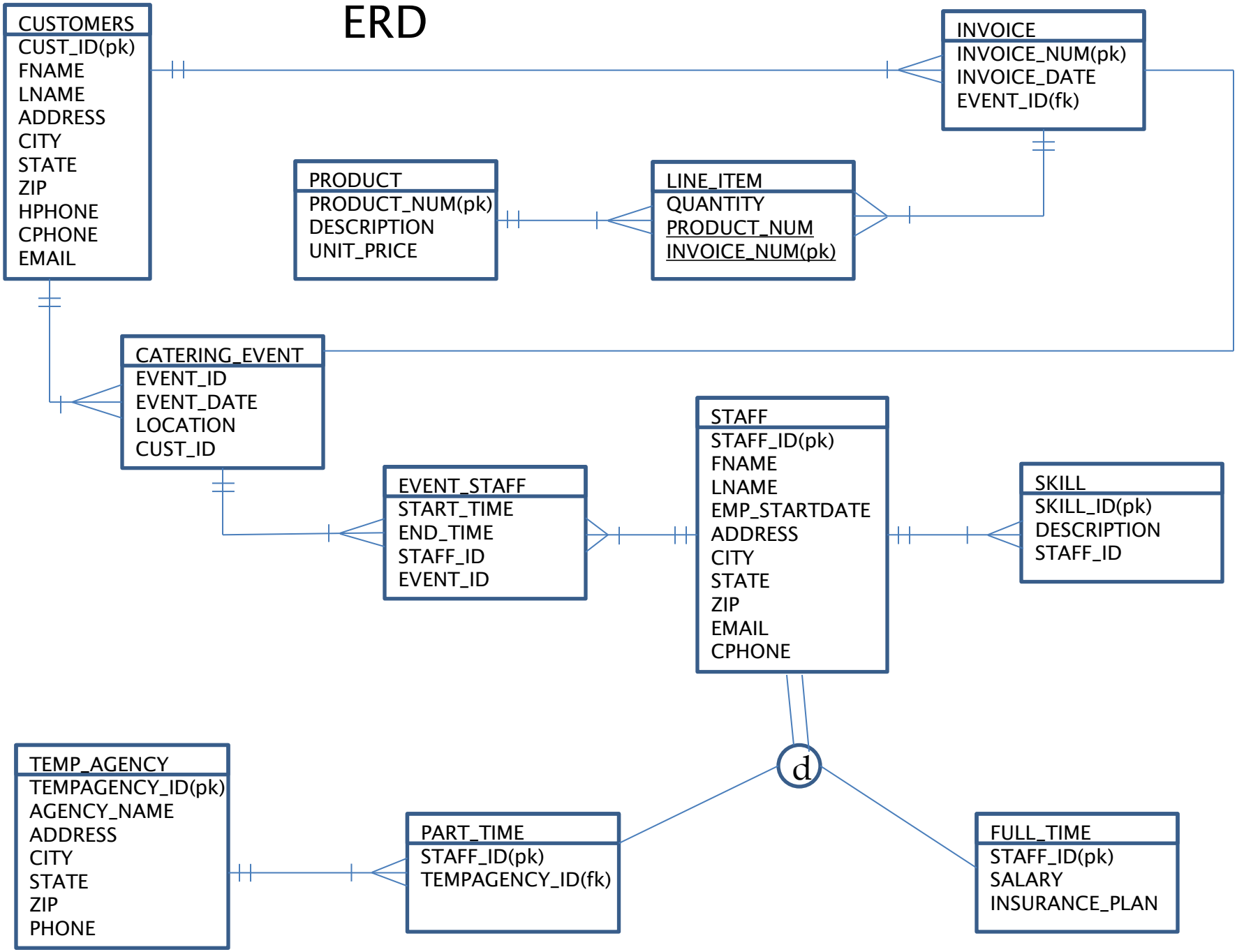
Ryan Ellorin

Business Rules Assumptions

To keep the Dates correct, the earliest that the Staff could begin working is 8am, and the latest is 10pm.

The catering business began on Jan 1, 2014 and the sample data ran until May 1st, 2014.

There was a total of 10 Staff, with each working random events.


ERD
CUSTOMERS
CUST_ID(pk)
FNAME
LNAME
ADDRESS
CITY
STATE
ZIP
HPHONE
CPHONE
EMAIL
INVOICE
INVOICE_NUM(pk)
INVOICE_DATE
EVENT_ID(fk)
PRODUCT
PRODUCT_NUM(pk)
DESCRIPTION
UNIT_PRICE
LINE_ITEM
QUANTITY
PRODUCT_NUM
INVOICE_NUM(pk)
CATERING_EVENT
EVENT_ID
EVENT_DATE
LOCATION
CUST_ID
EVENT_STAFF
START_TIME
END_TIME
STAFF_ID
EVENT_ID
STAFF
STAFF_ID(pk)
FNAME
LNAME
EMP_STARTDATE
ADDRESS
CITY
STATE
ZIP
EMAIL
CPHONE
SKILL
SKILL_ID(pk)
DESCRIPTION
STAFF_ID
d
TEMP_AGENCY
TEMPAGENCY_ID(pk)
AGENCY_NAME
ADDRESS
CITY
STATE
ZIP
PHONE
PART_TIME
STAFF_ID(pk)
TEMPAGENCY_ID(fk)
FULL_TIME
STAFF_ID(pk)
SALARY
INSURANCE_PLAN

Relational Schema

CUSTOMERS
CUST_ID (pk) FNAME, LNAME, ADDRESS, CITY, STATE, ZIP, HPHONE, CPHONE, EMAIL

CATERING_EVENT
EVENT_ID (pk), EVENT_DATE, LOCATION, CUST_ID(fk)

INVOICE
INVOICE_NUM (pk), INVOICE_DATE, EVENT_ID (fk)

PRODUCT
PRODUCT_NUM (pk), DESCRIPTION, UNIT_PRICE

LINE_ITEM
QUANTITY PRODUCT_NUM, INVOICE_NUM(pk),

STAFF
STAFF_ID (pk), FNAME, LNAME, EMP_STARTDATE, ADDRESS, CITY, STATE, ZIP, EMAIL CELLPHONE

SKILL
SKILL_ID (pk), DESCRIPTION, STAFF_ID(fk)

TEMP_AGENCY
TEMPAGENCY_ID (pk), AGENCY_NAME, ADDRESS, CITY, STATE, ZIP, PHONE

PART_TIME
STAFF_ID (pk), TEMPAGENCY_ID(fk)

FULL_TIME
STAFF_ID (pk), SALARY, INSURANCE_PLAN

EVENT_STAFF
START_TIME, END_TIME, STAFF_ID, EVENT_ID(pk)

Lessons Learned

Connecting the Date datatype for START_TIME and END_TIME for each Staff Member to coincide with the Date of the Catering Event requires correct, corresponding, and specific sample data. Even though I can still query the # of hours that each Staff member worked for a specific event, the date of the specific event do not match up to the date of the start time. I probably should have used Date as a PK.

I made multiple changes to my ERD as I was building my database and coming across problems (like the one above). If I spent more time on making my ERD and analyzing the data, I would've had less time altering tables and constraints.

Again, I found sample data to be VERY IMPORTANT as I had to redo the data for Invoice Numbers to coincide with Event Dates. It didn't make sense for a customer to be billed before the event.

Sample Data Information

I took product information from a local catering business

http://www.coastalcateringcompany.com/

and used generatedata.com to generate sample data such as names, addresses, city, states, zip, phone numbers, email, and event locations.

Create Table Statements

```
create table CUSTOMERS(
                CUST_ID varchar(10) not null,
                FNAME varchar(25) not null,
                LNAME varchar(25) not null,
                ADDRESS varchar(50),
                CITY varchar(25),
                STATE char(2),
                ZIP char(5),
                HPHONE char(14),
                CPHONE char(14),
                EMAIL varchar(50),
                CONSTRAINT CUSTOMERS_pk PRIMARY KEY (CUST_ID)
);

create table CATERING_EVENT(
                EVENT_ID varchar(10) not null,
                EVENT_DATE date,
                LOCATION varchar(50),
                CUST_ID varchar(10), --FK
                CONSTRAINT CATERING_EVENT_pk PRIMARY KEY (EVENT_ID),
                CONSTRAINT CATERING_EVENT_Custfk
                                FOREIGN KEY (CUST_ID)
                                REFERENCES CUSTOMERS(CUST_ID)
                                ON DELETE SET NULL;
);

create table INVOICE(
                INVOICE_NUM varchar(10) not null,
                INVOICE_DATE date not null,
                EVENT_ID varchar(10),
                CONSTRAINT INVOICE_pk PRIMARY KEY  (INVOICE_NUM,EVENT_ID),
                CONSTRAINT INVOICE_Eventfk
                                FOREIGN KEY (EVENT_ID)
                                REFERENCES CATERING_EVENT(EVENT_ID)

);
```

```
create table PRODUCT(
                PRODUCT_NUM varchar(10) not null,
                DESCRIPTION varchar(50) not null,
                UNIT_PRICE float,
                CONSTRAINT PRODUCT_pk PRIMARY KEY
(PRODUCT_NUM)
                CONSTRAINT Product_Invoicefk
                                FOREIGN KEY (INVOICE_NUM)
                                REFERENCES INVOICE(INVOICE_NUM)
                                ON DELETE SET NULL;
);

create table LINE_ITEM(
                QUANTITY number(10) not null,
                PRODUCT_NUM varchar(10),
                INVOICE_NUM varchar(10),
                CONSTRAINT LINE_ITEM_pk PRIMARY KEY
(PRODUCT_NUM,INVOICE_NUM),
);

create table STAFF(
                STAFF_ID varchar(10) not null,
                FNAME varchar(15),
                LNAME varchar(15),
                EMP_STARTDATE date,
                ADDRESS varchar(30),
                CITY varchar(15),
                STATE char(2),
                ZIP varchar(5),
                EMAIL varchar(50),
                CPHONE varchar(15),
                CONSTRAINT STAFF_pk PRIMARY KEY (STAFF_ID)
);

create table SKILL(
                SKILL_ID varchar(4) not null,
                DESCRIPTION varchar(50),
                STAFF_ID varchar(10),
                CONSTRAINT SKILL_pk PRIMARY KEY
(SKILL_ID,STAFF_ID),
                CONSTRAINT SKILL_STAFFfk
                                FOREIGN KEY (STAFF_ID)
                                REFERENCES STAFF(STAFF_ID)
                                ON DELETE SET NULL
);
```

Create Table Statements

```
create table TEMP_AGENCY(
                    TEMPAGENCY_ID varchar(10) not null,
                    AGENCY_NAME varchar(30) not null,
                    ADDRESS varchar(50),
                    CITY varchar(30),
                    STATE char(2),
                    ZIP  varchar(5),
                    PHONE varchar(15),
                    CONSTRAINT TEMP_AGENCY_pk PRIMARY KEY (TEMPAGENCY_ID)
);

create table PART_TIME(
                    STAFF_ID varchar(10) not null,
                    TEMPAGENCY_ID varchar(10),
                    CONSTRAINT PART_TIME_pk PRIMARY KEY (STAFF_ID),
                    CONSTRAINT PART_TIME_fk
                                FOREIGN KEY (TEMPAGENCY_ID)
                                REFERENCES TEMP_AGENCY(TEMPAGENCY_ID)
);

create table FULL_TIME(
                    STAFF_ID varchar(10) not null,
                    SALARY varchar(10),
                    INSURANCE_PLAN varchar(15),
                    CONSTRAINT FULL_TIME_pk PRIMARY KEY (STAFF_ID)
);

create table EVENT_STAFF(
                    START_TIME timestamp(6),
                    END_TIME timestamp(6),
                    STAFF_ID varchar(10),
                    EVENT_ID varchar(10),
                    CONSTRAINT EVENTSTAFF_PK PRIMARY KEY (STAFF_ID,EVENT_ID)
);
```

Describe Table Statements

Describe customers;

Name	Null?	Type
CUST_ID	NOT NULL	VARCHAR2(10)
FNAME	NOT NULL	VARCHAR2(25)
LNAME	NOT NULL	VARCHAR2(25)
ADDRESS		VARCHAR2(50)
CITY		VARCHAR2(25)
STATE		CHAR(2)
ZIP		CHAR(5)
HPHONE		VARCHAR2(14)
CPHONE		VARCHAR2(14)
EMAIL		VARCHAR2(50)

Describe Catering_Event;

Name	Null?	Type
EVENT_ID	NOT NULL	VARCHAR2(10)
EVENT_DATE		DATE
LOCATION		VARCHAR2(50)
CUST_ID		VARCHAR2(10)

Describe Invoice;

Name	Null?	Type
INVOICE_NUM	NOT NULL	VARCHAR2(10)
INVOICE_DATE	NOT NULL	DATE
EVENT_ID	NOT NULL	VARCHAR2(10)

Describe Table Statements

Describe Product;

Name	Null?	Type
PRODUCT_NUM	NOT NULL	VARCHAR2(10)
DESCRIPTION	NOT NULL	VARCHAR2(300)
UNIT_PRICE		FLOAT(126)

Describe Line_Item;

Name	Null?	Type
QUANTITY	NOT NULL	NUMBER(10)
PRODUCT_NUM	NOT NULL	VARCHAR2(10)
INVOICE_NUM	NOT NULL	VARCHAR2(10)

Describe Staff;

Name	Null?	Type
STAFF_ID	NOT NULL	VARCHAR2(10)
FNAME		VARCHAR2(15)
LNAME		VARCHAR2(15)
EMP_STARTDATE		DATE
ADDRESS		VARCHAR2(30)
CITY		VARCHAR2(15)
STATE		CHAR(2)
ZIP		VARCHAR2(5)
EMAIL		VARCHAR2(50)
CPHONE		VARCHAR2(15)

Describe Table Statements

Describe Skill;

Name	Null?	Type
SKILL_ID	NOT NULL	VARCHAR2(4)
DESCRIPTION		VARCHAR2(50)
STAFF_ID	NOT NULL	VARCHAR2(10)

Describe Temp_Agency;

Name	Null?	Type
TEMPAGENCY_ID	NOT NULL	VARCHAR2(10)
AGENCY_NAME	NOT NULL	VARCHAR2(30)
ADDRESS		VARCHAR2(50)
CITY		VARCHAR2(30)
STATE		CHAR(2)
ZIP		VARCHAR2(5)
PHONE		VARCHAR2(15)

Describe Part_Time;

Name	Null?	Type
STAFF_ID	NOT NULL	VARCHAR2(10)
TEMPAGENCY_ID		VARCHAR2(10)

Describe Full_Time;

Name	Null?	Type
STAFF_ID	NOT NULL	VARCHAR2(10)
SALARY		VARCHAR2(10)
INSURANCE_PLAN		VARCHAR2(15)

Describe Table Statements

Describe Event_staff;

Name	Null?	Type
START_TIME		DATE
END_TIME		DATE
STAFF_ID	NOT NULL	VARCHAR2(10)
EVENT_ID	NOT NULL	VARCHAR2(10)

Queries

Query specifying specific columns, using a column alias

List the total hours that STAFF_ID 2001 worked for

```
SELECT
(((END_TIME - START_TIME)*86400)/60)/60 "WORKED HOURS FOR STAFF_ID 2001" FROM EVENT_STAFF
WHERE STAFF_ID=2001;
```

WORKED HOURS FOR STAFF_ID 2001
6
5
6
3
6
6
4
6
4
3
4
3
6
4

Query using the concatenation operator

List the Company Staff First and Last names

```
select fname||' '||lname "Staff“ from staff;
```

Staff
Sydney Allen
Roth Sullivan
Cassandra Klein
Quentin Byers
Tanek Mcgee
Charles Burns
Pamela Buck
Simone Head
Rhoda Swanson
Elvis Espinoza

Queries

Query using a literal character string

Find out when Sydney started working

```
SELECT CONCAT(CONCAT(FNAME,' started working here on '),EMP_STARTDATE) "STAFF_ID 2000 Start Date"
   FROM STAFF WHERE staff_id = 2000;
```

STAFF_ID 2000 Start Date
Sydney started working here on 25-MAR-14

Query using the where clause and a comparison operators

Find which Invoice sale numbers total more than $150

```
SELECT INVOICE_NUM "Invoice# where Total Sale>$150“
FROM LINE_ITEM INNER JOIN PRODUCT ON LINE_ITEM.PRODUCT_NUM = PRODUCT.PRODUCT_NUM
WHERE LINE_ITEM.QUANTITY*PRODUCT.UNIT_PRICE > 150.00;
```

Invoice# where Total Sale>$150
100004
100005
100006
100007
100009
100011
100012
100017
100029
100030

Queries

Query using the BETWEEN operator

Find which staff members began working in February 2014

```
SELECT *
FROM staff
WHERE emp_startdate BETWEEN TO_DATE ('2014/02/01', 'yyyy/mm/dd')
AND TO_DATE ('2014/02/28', 'yyyy/mm/dd');
```

STAFF_ID	FNAME	LNAME	EMP_STARTDATE	ADDRESS	CITY	STATE	ZIP	EMAIL	CPHONE
2003	Quentin	Byers	28-FEB-14	P.O. Box 680, 7706 Elit, St.	Savannah	GA	74522	tempor@Duis.net	1-991-849-8796
2004	Tanek	Mcgee	23-FEB-14	P.O. Box 502, 9313 Aliquet Av.	Denver	CO	40670	Phasellus.dapibus.quam@justonec.net	1-465-128-8466
2007	Simone	Head	25-FEB-14	P.O. Box 326, 3428 Ipsum Rd.	Casper	WY	72819	lectus.pede@hendreritconsectetuercursus.edu	1-220-873-7361

Queries

Query using the IN(LIST) operator

Find which customers have the first name Nero, Rhoda, or Ivory

```
SELECT * FROM CUSTOMERS
WHERE FNAME IN ('Nero','Rhoda','Ivory');
```

CUST_ID	FNAME	LNAME	ADDRESS	CITY	STATE	ZIP	HPHONE	CPHONE	EMAIL
7	Rhoda	Solis	650-366 Dolor Rd.	Huntsville	AL	35293	1-664-482-0559	1-568-344-3741	tempor.arcu@fames.ca
17	Nero	Mcbride	910-8980 Ut Street	Grand Island	NE	62232	1-333-276-2346	1-869-594-0196	nec.urna@pellentesqueSed.com
25	Ivory	Baird	P.O. Box 426, 3668 Purus Rd.	West Jordan	UT	97257	1-405-495-0990	1-378-434-7142	nibh.vulputate@quamelementumat.edu

Query using the LIKE operator

Find which customers have the last name that starts with “BA”

```
SELECT LNAME FROM customers WHERE LNAME LIKE 'Ba%';
```

LNAME
Ballard
Baird

Queries

Query using the AND, OR, ORDER BY, LPAD() function

Find which staff members are from LA with the first name Roth OR from Phoenix, Order by Ascending Staff_ID, Use first initial of first name

```
select Lpad(Fname,1),Lname,Address,City,State,staff_id from staff
where STATE='LA' and FNAME='Roth'
or City like'Ph%'
ORDER BY STAFF_ID ASC;
```

LPA	LNAME	ADDRESS	CITY	STATE	STAFF_ID
R	Sullivan	P.O. Box 326, 328 Sit Av.	Metairie	LA	2001
P	Buck	Ap #420-729 Ridiculus Rd.	Phoenix	AZ	2006
E	Espinoza	8559 Euismod Rd.	Phoenix	AZ	2009

Queries

Query using the NOT, LOWER() function

Find which staff members doesn't begin with S; names in lowercase

```
SELECT lower(FNAME)
FROM staff
WHERE fname NOT LIKE 'S%';
```

LOWER(FNAME)
roth
cassandra
quentin
tanek
charles
pamela
rhoda
elvis

Query using LEFT OUTER JOIN, UPPER() function

Match Part time Staff ID's with Temp Agency's, in Uppercase

```
select part_time.staff_id, upper(temp_agency.agency_name)
from part_time
left outer join temp_agency
on temp_agency.tempagency_id = part_time.tempagency_id;
```

STAFF_ID	UPPER(TEMP_AGENCY.AGENCY_NAME)
2005	EU SEM LIMITED
2006	MASSA MAURIS FOUNDATION
2001	MASSA MAURIS FOUNDATION
2003	AT LTD
2009	SED DICTUM ELEIFEND INC.
2004	SED DICTUM ELEIFEND INC.
2007	EU TELLUS COMPANY

Queries

Query using Inner join, Order by column alias, Ascending

List Staff with their Skills

```
SELECT s.fname, s.lname as "LAST NAME", k.description
FROM staff s
inner join skill k
ON s.staff_id = k.staff_id
ORDER BY LNAME asc;
```

FNAME	LAST NAME	DESCRIPTION
Sydney	Allen	MANAGER
Sydney	Allen	CHEF
Pamela	Buck	SERVER
Charles	Burns	SERVER
Quentin	Byers	SET UP STAFF
Quentin	Byers	DELIVERER
Elvis	Espinoza	SET UP STAFF
Elvis	Espinoza	SERVER
Simone	Head	SERVER
Cassandra	Klein	MANAGER
Cassandra	Klein	BAKER
Tanek	Mcgee	SERVER
Tanek	Mcgee	SET UP STAFF
Roth	Sullivan	MANAGER ASSISTANT
Rhoda	Swanson	CHEF

Queries

Query using SUBSTR()

Select product where unit price = 8.49

```
select SUBSTR( description, 1,6 ) from product
where unit_price = 8.49;
```

SUBSTR(DESCRIPTION
Lunch
Lunch
Lunch
Lunch
Lunch
Lunch
Lunch

Query using INSTR()

Select product where unit price = 8.49

```
select INSTR( description, 'c' ) from product
where unit_price = 8.49;
```

INSTR(DESCRIPTION,'C')
5
5
5
5
5
5
5

Queries

Query using SELF JOIN, RPAD()

Select Staff members from AZ

```
SELECT rpad(F.STAFF_ID,10,'X'),F.LNAME, S.LNAME
FROM STAFF F, STAFF S
WHERE F.STATE = S.STATE
AND F.STAFF_ID < S.STAFF_ID;
```

RPAD(F.STAFF_ID,10,'X')	LNAME	LNAME
2001XXXXXX	Sullivan	Burns
2006XXXXXX	Buck	Espinoza

Query using MONTHS_BETWEEN

Staff Employment Start date to May 1, 2014

```
SELECT MONTHS_BETWEEN (TO_DATE (EMP_STARTDATE, 'yyyy/mm/dd'), TO_DATE ('2014/05/01', 'yyyy/mm/dd') ) "Employment Length"
FROM STAFF;
```

Employment Length
-23869.581
-24039.581
-24075.581
-23834.581
-23894.581
-23823.581
-24073.581
-23870.581
-23797.581
-23893.581

Queries

Query using HAVING, GROUP BY, and COUNT()

Select Skill positions that have more than one staff member, and display it

```
SELECT description, COUNT(*)
FROM skill
WHERE staff_id > 2000
GROUP BY description
HAVING COUNT(*)>1;
```

DESCRIPTION	COUNT(*)
SET UP STAFF	3
SERVER	5