**Giving you ERD for Jeff’s Shop answer the following questions:**

1. Draw a complete ERD for your solution showing all tables.
2. Select all customers with or without orders, for customers with orders list all order information

--Solution:

**SELECT** c.\*, o.\*

**FROM** (

**SELECT** first**\_**name, last\_name, customer\_id

**FROM** tbl\_customers) c

**FULL** **OUTER** **JOIN** tbl\_orders o

**ON** c.customer\_id = o.customer\_id

**WHERE** c.customer\_id **IS** **NOT** **NULL**;

1. List customers with orders only.

--Solutoin 1:

**SELECT** \*

**FROM** tbl\_customers

**WHERE** **EXISTS** (

**SELECT** order\_id

**FROM** tbl\_orders

**WHERE** tbl\_customers.customer\_id = tbl\_orders.customer\_id);

--Solution 2:

**SELECT** \*

**FROM** tbl\_customers

**WHERE** customer\_id **IN** (

**SELECT** customer\_id

**FROM** tbl\_orders);

1. Find a Cartesianproduct of the customer and order tables

--Solution:

**SELECT** c.\*, o.\*

**FROM** tbl\_customers c,

(**SELECT** \* **FROM** tbl\_orders) o;

1. List all items that have not been ordered yet.

--Solution:

**SELECT** part\_number,

part\_name

**FROM** tbl\_parts

**WHERE** part\_number **NOT** **IN** (

**SELECT** part\_number

**FROM** tbl\_part\_details);

1. List all ordered items

--Solution:

**SELECT** part\_number,

part\_name

**FROM** tbl\_parts

**WHERE** part\_number **IN** (

**SELECT** part\_number

**FROM** tbl\_part\_details);

1. List all orders that the customer did notpay for.

--Solution:

**SELECT** \*

**FROM** tbl\_orders

**WHERE** order\_id **IN** (

**SELECT** order\_id

**FROM** tbl\_payments

**WHERE** paid\_amount = 0);

1. List all orders paid with a check (assume you have a separate payment table)

--Solution:

**SELECT** o.\*

**FROM** tbl\_orders o

**RIGHT** **OUTER** **JOIN** tbl\_payments p

**ON** o.order\_id = p.order\_id

**WHERE** p.method\_id **IN** (

**SELECT** method\_id

**FROM** tbl\_payments

**NATURAL** **JOIN** tbl\_payment\_methods

**WHERE** method\_name = 'check');

1. Give all ordered items in Jan 2014 and not in Jan 2013.

--Solution:

**SELECT** \*

**FROM** tbl\_parts

**WHERE** part\_number **IN** (

**SELECT** part\_number

**FROM** tbl\_part\_details

**WHERE** order\_id **IN** (

**SELECT** order\_id

**FROM** tbl\_orders

**WHERE** order\_time>= to\_date('01-01-2014','dd-mm-yyyy')

**AND** order\_time<= to\_date('31-01-2014','dd-mm-yyyy'))

);

1. List customer names and phone numbers from Texas with orders over $1000.

--Solution:

**SELECT** \*

**FROM** tbl\_customers

**WHERE** customer\_id **IN** (

**SELECT** customer\_id

**FROM** tbl\_customer\_address

**WHERE** tbl\_customer\_address.state = 'TX')

**AND** customer\_id **IN** (

**SELECT** customer\_id

**FROM** tbl\_orders

**WHERE** tbl\_orders.price> 1000);

1. List all customers names along with their cities weather the customer has a city or not.

--Solution:

**SELECT** c.\*, a.city

**FROM** (**SELECT** first\_name, last\_name, customer\_id

**FROM** tbl\_customers) c

**FULL** **OUTER** **JOIN** tbl\_customer\_address a

**ON** c.customer\_id = a.customer\_id

**WHERE** c.customer\_id **IS** **NOT** **NULL**;

1. List all customers with cancelled orders.

--Solution:

**SELECT** first\_name, last\_name

**FROM** tbl\_customers

**WHERE** customer\_id **IN** (

**SELECT** customer\_id

**FROM** tbl\_orders

**WHERE** order\_status = 'cancell');

1. List all vendors and their items (list vendors only who has items in the database. Assume we have a separate table for vendors)

--Solution:

**SELECT** supplier\_id, company\_name, part\_name

**FROM** tbl\_suppliers

**JOIN** tbl\_supply\_details using (supplier\_id)

**JOIN** tbl\_parts using (part\_number)

**WHERE** supplier\_id **IN** (

**SELECT** supplier\_id

**FROM** tbl\_supply\_details);

1. List sales total per month

**SELECT** sum(price) totle, **TO\_CHAR**(order\_time, 'YYYY') year, **TO\_CHAR**(order\_time, 'MONTH') month

**FROM** tbl\_orders

**GROUP** **BY** **TO\_CHAR**(order\_time, 'YYYY'), **TO\_CHAR**(order\_time, 'MONTH')

1. List the customer with the highest order total

--Solution:

**SELECT** c.\*, o.total

**FROM** tbl\_customers c

**INNER** **JOIN** (

**SELECT** customer\_id,

**SUM**(price) **AS** total

**FROM** tbl\_orders

**GROUP** **BY** customer\_id) o

**ON** c.customer\_id = o.customer\_id

**WHERE** o.total = (

**SELECT** **MAX**(**SUM**(price))

**FROM** tbl\_orders

**GROUP** **BY** customer\_id);

1. List the address and name of the customers who placed at least 2 orders

--Solution:

**SELECT** c.first\_name, c.last\_name, a.street\_address, a.postal\_code, a.city,a.state, a.address\_type, o.order\_numbers

**FROM** tbl\_customer\_address a

**INNER** **JOIN** tbl\_customers c

**ON** a.customer\_id = c.customer\_id

**JOIN** (

**SELECT** **COUNT**(order\_id) **AS** order\_numbers, customer\_id

**FROM** tbl\_orders

**GROUP** **BY** customer\_id

**ORDER** **BY** customer\_id) o

**ON** c.customer\_id = o.customer\_id

**WHERE** order\_numbers>= 2;

1. List the average order total for all customers from Chicago

--Solution 1:

**SELECT** c.first\_name, c.last\_name, a.street\_address, a.postal\_code, a.city, a.state, o.average\_price

**FROM** tbl\_customer\_address a

**INNER** **JOIN** tbl\_customers c

**ON** a.customer\_id = c.customer\_id

**JOIN** (

**SELECT** customer\_id,

**AVG**(price) **AS** average\_price

**FROM** tbl\_orders

**GROUP** **BY** customer\_id

**ORDER** **BY** customer\_id) o

**ON** c.customer\_id = o.customer\_id

**WHERE** a.city = 'Chicago';

--Solution 2:

**SELECT** avg(price)

**FROM** tbl\_orders

**WHERE** address\_id **IN**(

**SELECT** address\_id

**FROM** tbl\_customer\_address

**WHERE** city = 'Chicago');

1. For each item list customers’ names who ordered it

**SELECT** part\_number, part\_name, first\_name, last\_name

**FROM** tbl\_parts

**JOIN** tbl\_part\_details **USING** (part\_number)

**JOIN** tbl\_orders **USING** (order\_id)

**JOIN** tbl\_customers **USING** (customer\_id);

1. List the number of orders for each state (customer’s state).

**SELECT** a.state, **COUNT**(o.order\_id) **AS** order\_numbers

**FROM** tbl\_customer\_address a

**LEFT** **OUTER** **JOIN** tbl\_orders o

**ON** o.address\_id = a.address\_id

**GROUP** **BY** a.state

**ORDER** **BY** a.state;

1. List the name of your favorite professor.

**SELECT** professor\_name

**FROM** tbl\_professors

**WHERE** professor\_number =(

**SELECT** professor\_number

**FROM** class

**WHERE** class\_id = 422)