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

1. **Draw a complete ERD for your solution showing all tables.**

Please see the file of final version of ERD for small engine shop.vsdx.

1. **Select all customers with or without orders, for customers with orders list all order information**

**--Solutoin:**

SELECT first\_name,

last\_name,

tbl\_orders.\*

FROM tbl\_customers

LEFT OUTER JOIN tbl\_orders

ON tbl\_customers.customer\_id = tbl\_orders.customer\_id;

1. **List customers with orders only.**

**--Solution 1:**

SELECT first\_name,

last\_name,

tbl\_orders.\*

FROM tbl\_customers

INNER JOIN tbl\_orders

ON tbl\_customers.customer\_id = tbl\_orders.customer\_id;

**--Solution 2:**

SELECT first\_name,

last\_name,

tbl\_orders.\*

FROM tbl\_customers,

tbl\_orders

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

**--Solution 3:**

SELECT first\_name,

last\_name,

tbl\_orders.\*

FROM tbl\_customers

Right OUTER JOIN tbl\_orders

ON tbl\_customers.customer\_id = tbl\_orders.customer\_id;

1. **Find a Cartesian product of the customer and order tables**

**--Solution 1:**

SELECT \*

FROM tbl\_customers

CROSS JOIN tbl\_orders;

**--Solution 2:**

SELECT \*

FROM tbl\_customers, tbl\_orders;

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

**--Solution 1:**

SELECT part\_number,

part\_name

FROM tbl\_parts

WHERE part\_number NOT IN (

SELECT part\_number

FROM tbl\_part\_details);

**--Solution 2:**

SELECT p.part\_number,

p.part\_name,

d.order\_id

FROM tbl\_parts p

LEFT OUTER JOIN tbl\_part\_details d

ON p.part\_number = d.part\_number

WHERE d.order\_id IS NULL;

1. **List all ordered items**

**--Solution 1:**

SELECT part\_number,

part\_name

FROM tbl\_parts

WHERE part\_number IN (

SELECT part\_number

FROM tbl\_part\_details);

**--Solution 2:**

SELECT p.part\_number,

p.part\_name,

d.order\_id

FROM tbl\_parts p

LEFT OUTER JOIN tbl\_part\_details d

ON p.part\_number = d.part\_number

WHERE d.order\_id IS NOT NULL;

**--Solution 3:**

SELECT part\_number,

part\_name,

order\_id,

order\_time

FROM tbl\_orders

JOIN tbl\_part\_details USING (order\_id)

JOIN tbl\_parts USING (part\_number)

ORDER BY 1;

1. **List all orders that the customer did not pay for.**

**--Solution:**

SELECT o.\*,

p.paid\_amount

FROM tbl\_orders o

JOIN tbl\_payments p

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

WHERE p.paid\_amount = 0;

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

**--Solution 1:**

SELECT order\_id,

customer\_id,

order\_time,

price,

order\_status,

paid\_amount,

method\_name

FROM tbl\_orders

JOIN tbl\_payments USING (order\_id)

JOIN tbl\_payment\_methods USING (method\_id)

WHERE method\_name = 'check';

**--Solution 2:**

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 order\_id,

order\_time,

price,

part\_number,

part\_name

FROM tbl\_orders

JOIN tbl\_part\_details USING (order\_id)

JOIN tbl\_parts USING (part\_number)

WHERE TRUNC(tbl\_orders.order\_time)

BETWEEN '01-JAN-2014' AND '31-JAN-2014'

AND TRUNC(tbl\_orders.order\_time)

NOT BETWEEN '01-JAN-2013' AND '31-JAN-2013';

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

**--Solution:**

SELECT first\_name,

last\_name,

phone\_number

FROM tbl\_customers

JOIN tbl\_customer\_address USING (customer\_id)

JOIN tbl\_orders USING (address\_id)

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

AND tbl\_orders.price > 1000;

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

**--Solution:**

SELECT first\_name,

last\_name,

city

FROM tbl\_customers

JOIN tbl\_customer\_address USING (customer\_id);

1. **List all customers with cancelled orders.**

**--Solution 1:**

SELECT first\_name,

last\_name,

order\_id,

order\_time,

price,

order\_status

FROM tbl\_customers

JOIN tbl\_orders USING (customer\_id)

WHERE order\_status = 'cancell';

**--Solution 2:**

SELECT c.first\_name,

c.last\_name,

o.\*

FROM tbl\_customers c

RIGHT OUTER JOIN tbl\_orders o

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

WHERE o.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 company\_name,

contact\_name,

part\_number,

part\_name

FROM tbl\_suppliers

JOIN tbl\_supply\_details USING (supplier\_id)

JOIN tbl\_parts USING (part\_number);

1. **List sales total per month**

--Solution:

SELECT TO\_CHAR(order\_time, 'YYYY') year,

TO\_CHAR(order\_time, 'MONTH') month,

SUM(price) total\_sales

FROM tbl\_orders

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

TO\_CHAR(order\_time, 'MONTH'),

TO\_CHAR(order\_time, 'MM')

ORDER BY TO\_CHAR(order\_time, 'YYYY'),

TO\_CHAR(order\_time, 'MM');

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:**

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';

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

**--Solution:**

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).**

**--Solution:**

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.**

**--Solution:**

SELECT first\_name, last\_name

FROM course

WHERE course\_id = ‘ITM 422’