**DBS Lab 2**

**Name:** Tommy Umholtz

**Student Number:** 0004654267

**1. Perform a SQL Dump of your example database using Sequel Pro. Do not Upload it to FSO. This is for your backup purposes. (1pt)**

**2. Create a new user (as yourself or favorite movie character) by inserting into the users table. As the userId is a varchar, choose a random, but unique value. Be sure to populate ALL columns. Paste the SQL below. (1pt)**

|  |
| --- |
| insert into users  (userid, firstname, lastname, middlename, username, password, dob, gender, occupationId, userStatusId, userTypeId)  values ('abce', 'Tommy', 'Holtz','P','tholtz','123','1981-08-02','M','134','1','3'); |

**3. Create 2 new orders for the user created in question 2 by inserting into the orders and orderItem tables. Place at least 1 item per order. Paste the SQL below. (1pt)**

|  |
| --- |
| insert into orders (orderId, userId, orderDate, shippingDate) values ('21275', 'abce', '2016-03-08 24:11:57', '2016-03-11 23:11:57');  insert into orders (orderId, userId, orderDate, shippingDate) values ('21276', 'abce', '2016-03-11 23:11:57', '2016-03-11 23:11:57')  insert into orderItem (orderId, itemId, quantity) values ('21275', '39', '2');  insert into orderItem (orderId, itemId, quantity) values ('21276', '11', '3'); |

**4. Assign 2 DVDs to the user created in question 2. Paste the SQL below. (1pt)**

|  |
| --- |
| insert into userDVD (userdvdId, userId, dvdId) values ('1000001', 'abce', '893'), ('1000002', 'abce', '1616'); |

**5. Assign 2 vehicles to the user created in question 2. Paste the SQL below. (1pt)**

|  |
| --- |
| insert into userVehicle (userVehicleId, UserId, vehicleId) values ('25001', 'abce', '1930'), ('25002', 'abce', '10576'); |

**6. Write a SQL statement to return the users (userId, firstname, lastname) and their orders (orderId), only for users that have placed an order. Join the users and orders tables. Paste the SQL below. (2pt)**

|  |
| --- |
| select users.userId, firstname, lastname, orderId from users  right join orders on users.userId = orders.userId;  or  select users.userId, firstname, lastname, orderId from users  join orders on users.userId = orders.userId  where orderId is not NULL; |

**7. Write a SQL statement to return all users (limit 200) and orders if available. When joining the 2 tables, be sure to choose the correct type of join. Paste SQL below. (2pt)**

|  |
| --- |
| select \* from users  left join orders on users.userId = orders.userId  limit 200; |

**8. Write a SQL statement to return all users (userId, firstname, lastname) who have not placed an order. Again, when joining the users and order tables, choose the correct join. (limit 200) Paste SQL below. (2pt)**

|  |
| --- |
| select users.userId, firstname, lastname from users  left join orders on users.userId = orders.userId  where orderId is NULL  limit 200; |

**9. Write a SQL statement to return all orders (orderId, orderDate), the information of the user who placed the order (userId, firstName, Lastname), and their items ordered and quantities (itemName, quantity). (limit 200)Paste the SQL below. (2pt)**

|  |
| --- |
| select orders.orderId, orderDate, users.userId, firstName, Lastname, itemName, quantity from orders  left join users on users.userId = orders.userId  left join orderItem on orderItem.orderId = orders.orderId  left join item on orderItem.itemId = item.itemId  Limit 200; |

**10. Alter the order table to add a column called “destinationState”. Paste the alter table SQL statement below.**

**Then populate the destinationState column for the existing orders.**

**Ex: update orders set destinationState = ‘FL’;**

**(2pt)**

|  |
| --- |
| ALTER TABLE ordersADD COLUMN destinationState VARCHAR (2) NULL AFTER shippingDate;  update orders set destinationState = 'FL'; |

**11. Create a foreign key between the tables, state and orders. IE: From table states, column state to table order table, column destinationState. Foreign Key (FK) always point at a Foreign Key (FK). Paste SQL (alter table) below. (2pt)**

|  |
| --- |
| ALTER TABLE orders  ADD CONSTRAINT FK\_orders  FOREIGN KEY (destinationState) REFERENCES state(state); |

**12. Select all users who have a lastname of “Steady” and display their full name as a single field. Paste SQL below and the time the query ran. (2pt)**

|  |
| --- |
| select concat (firstname, ' ', lastname) from users  where lastname = "Steady";  68.8 ms |

**13. Create an index on the last name column for the users table. (A regular index, not unique) Paste SQL below. (2pt)**

|  |
| --- |
| CREATE INDEX lastname  ON users (lastname); |

**14. Rerun the query from question 12. Paste SQL below and the time the query ran. (1pt)**

|  |
| --- |
| 3.2 ms |

**15. Get the count of all users with a last name ending in “ing”. Paste SQL and results below. (2pt)**

|  |
| --- |
| select concat (firstname, ' ', lastname) from users  where lastname = "Steady";  .5 ms |

**16. Get the count of all users with a first name starting with “J” and who are employed. Users who do not have an occupation are considered unemployed. Paste SQL and results below. (2pt)**

|  |
| --- |
| You needed to filter out the employed.  select count(occupationId) as count from users  where firstname like 'J%'  and occupationId is not null;  18487 |

**17. Create an ER diagram of the database using MySQL Workbench. Export the diagram as an image. (2pt)**

To complete this assignment, upload this word doc and your ER Diagram image to FSO.