Fall | 16

Database Project

Jacob R Aylward CS440F142

COSC 440

There is a new Car Rental Company going to start their business in 2015. You are employed to creating a DBMS for this company.

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Table of Contents

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 1: Business Requirement (Description and assumption of your project. This is also called Project Specification)

The business organization provided for this database project is as follows:
Branch (Branch ID *(range from 1,000 to 99,999)*, Name, Address, Phone Number, Number of Employees in that branch, Category *(Based on how many cars are assigned to the branch)*, Manager Name)
Employee (Emp ID, First Name, Last Name, Middle Initial, Address, SSN, Type *(Manager, Mechanic, Secretary, Sales Person, Web Manager)*, Yearly Salary, Phone ext, Birth Date, Date Hired, Sex, Race, Home Phone)
Customer (SSN, ID *(Range from 10,000 to 999,999)*, Name, Age, Birth date, Home Phone, Work Phone, Street Address, City, State, Zip Code, Email, Acct_ID, Driver's License Number, Driver's License State)
Vehicle (Car_ID, Registration *(Plate No, State registered, year)*, Class *(Compact, Economy, Luxury, Pickup, Van)*, Features *(2 doors, 4 doors)*, Make *(Chevy, Pontiac, Ford, Toyota, Honda, Mazda)*, Color, Original Price, Last Oil Change Date, Last Oil Change Mileage, Location of Last Oil Change, Last Service, Insurance No., VIN Number)
Transaction (Trans_ID, Start Date *(& time)*, End Date *(& time)*, Description, Miles Used During This Transaction, Price/Day, Tax, Insurance *(Additional insurance $)*, Balance *(total)*, Branch where car has been rented)

It also provides for us a set of detailed descriptions and limitations:
1. The car rental company has several branches
2. A branch hires employees
3. Each employee works for only one branch
4. Each branch has one manager
5. Manager is also an employee
6. Branch categories are less than 10, 11-40, 41-100, more than 100
7. A customer can receive 250 miles free and will be charged 0.41 for each additional mile when renting a car (per day)
8. An employee rents a car to a customer
9. A transaction is posted when a customer rents a car from an employee
10. Transaction is updated when the car is returned

After evaluating the details provided to us for the Database Project we are capable of making the following initial assumptions on how to go about setting up our database. The First thing provided to us for making our Database are the following entities: Branch, Employee, Customer, Vehicle and Transaction. When

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

going through our first entity set (Branch) we see the following attributes: Branch ID (Range from 1,000 to 999,999), Name, Address, Phone Number, Number of Employees in that branch, Category (Based on how many cars are assigned to the branch) and finally the Manager Name. The first Attribute (Branch ID) has a range between 1,000 - 99,999 therefore we can conclude that there is no constant value for this attribute. The next assumption I can make for this entity set is that the Address attribute will cause problems unless I break it down or make it type: TEXT. Finally the Category attribute at first glance looks like a calculation therefore I would most likely make this attribute a method or use COUNT.

The next entity set (Employee) has the following attributes: Emp ID, First Name, Last Name, Middle Initial, Address, SSN, Type (Manager, Mechanic, Secretary, Sales Person, Web Manager), Yearly Salary, Phone ext, Birth Date, Date Hired, Sex, Race and Home Phone. The first issue I can see with the available given attributes is type, the term type is already used as a function therefore cannot be used as a given name for an attribute. While on the subject of Type, the 'type' for this attribute once renamed would most likely be ENUM due to the given job types suggested. As said before the attribute Address could possibly become problematic later, therefore I should probably make a separate Entity specifically for storing all addresses under an ID. Yearly salary could become a personal problem for me when attempting to figure out how to set up the attribute type, because I still haven't fully grasped how to use the decimal data type. As far as Primary Keys, the EMP_ID is atomic as well as the SSN therefore either could be used.

In the Customer entity set, which contains the following attributes: SSN, ID (Range from 10,000 to 999,999), Name, Age, Birth date, Home Phone, Work Phone, Street Address, City, State, Zip Code, Email, Acct_ID, Driver's License Number and Driver's License State. We can make the following assumption based on the information provided: the first is that the ID for customers having a range from 10,000 to 999,999 means that the data type will have to invoke a CHECK that makes sure the data is BETWEEN those two individual sums. The next assumption that can be made for this entity set is that the Address is used to define the location of cars, branches, employees and customers; therefore it would be prominently easier to make Address into its own entity set and provide a primary key such as Address_ID to be used as a foreign key to all other entity's that require an address. That also means that attributes such as the 'Drivers_License_State' can call on Address_ID to store the information on individual states using a 'State_ID'.

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Following the Customer entity set is the Vehicle set which contains the following attributes provided: Car_ID, Registration (Plate No, State registered, year), Class (Compact, Economy, Luxury, Pickup, Van), Features (2 doors, 4 doors), Make (Chevy, Pontiac, Ford, Toyota, Honda, Mazda), Color, Original Price, Last Oil Change Date, Last Oil Change Mileage, Location of Last Oil Change, Last Service, Insurance No., Vin Number. The Entity set for Vehicle is too complicated, I think it would be more rational to break it into more than one entity. For example I would create a Car entity to handle the Car_ID, Class, Features, Make, color, and Total_Mileage. Then I would store the other attributes into Vehicle(Registration_State, Registration_Year, plate_No, Vin_No, Price, last_Oil_Change, Last_Oil_Change_Mileage and Last_Service_Date). The attribute for Insurance_No would be stored with customer seeing as later on in our documentation it ask us to provide a query for customers that have insurance.

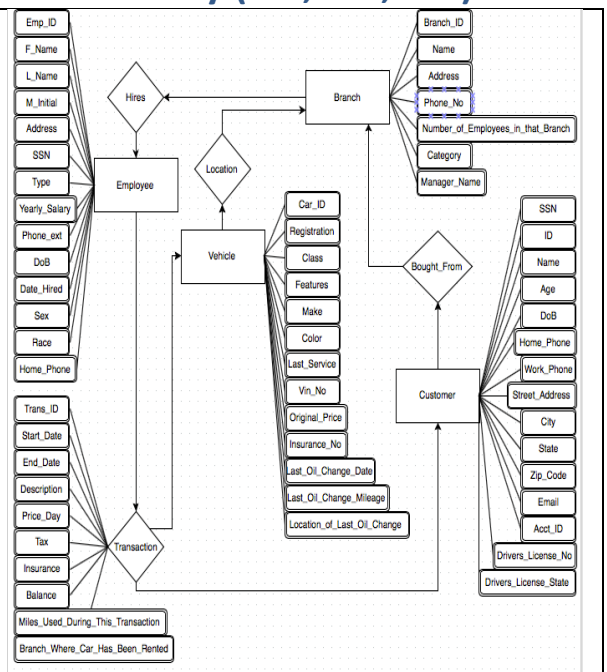
The Final Entity set that was provided for us is Transaction, but when looking at the attributes associated with this Entity set, I believe that the set is instead a relationship. The attributes for the Transaction relationship are as follows: Trans_ID, Start Date(& time), End Date (& time), Description, Miles used during this transaction , Price/Day, Tax, Insurance (Additional insurance $), Balance (total), Branch where car has been rented. From these attributes I see the that additional insurance is a probable problem due to the fact that this can be seen as a calculation. The only other thing that can be pointed out is the Description attribute, I will need to remember not to simplify the name of description into ‘desc’ or ‘describe’ due to the fact that these are alreading terms used in sql for functions.

1. The car rental company has several branches.
Ill need to create several inset statements to create individual branches.
2. A branch hires employees
Create a relationship named ‘Hires’ connecting branch to employees.
4. Each branch has one manager
Create/ insert one employee of manager type for each created branch.
5. Manager is also an employee
Manager is a type in the Employee Entity.
8. An employee rents a car to a customer
A log/attribute that states which EMP_ID rented out a car to Cust_ID.

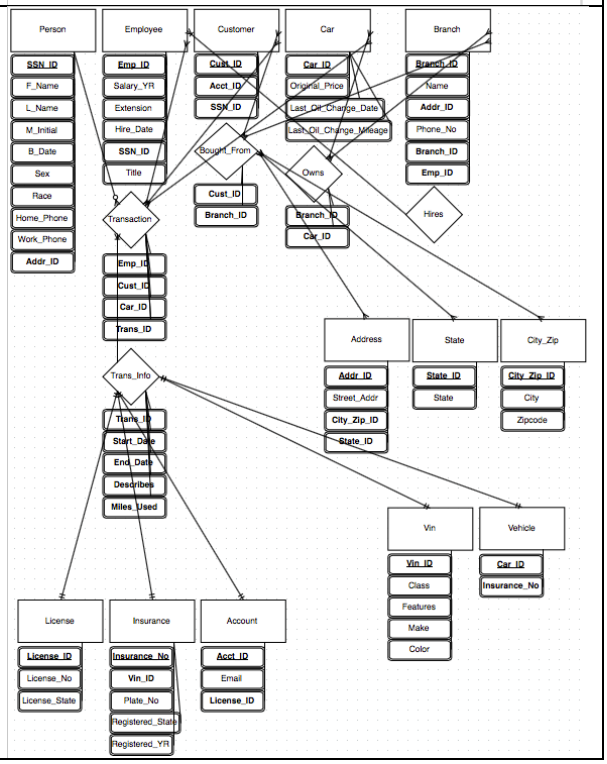
Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 2: Complete ERM with Functionality (1-1, 1-n, n-n)

First Database Design:
Branch(Branch_ID, Name, Address, Phone_No, Number_of_Employees_in_that_Branch, Category, Manager_Name)
Employee(Emp_ID, F_Name, L_Name, M_Initial, Address, SSN, Type, Yearly_Salary, Phone_ext, DoB, Date_Hired, Sex, Race, Home_Phone)
Vehicle(Car_ID, Registration, Class, Features, Make, Color, Last_Service, Vin_No, Original_Price, Insurance_No, Last_Oil_Change_Date, Last_Oil_Change_Mileage, Location_of_Last_Oil_Change)
Customer(SSN, ID, Name, Age, DoB, Home_Phone, Work_Phone, Street_Address, City, State, Zip_Code, Email, Acct_ID, Drivers_License_No, Drivers_License_State)
Hires()
Location()
Bought From()
Transaction(Trans_ID, Start_Date, End_Date, Description, Price_Day, Tax, Insurance, Balance, Miles_Used_During_This_Transaction, Branch_Where_Car_Has_Been_Rented)



Final Database Design:
Branch(**Branch_ID**, Name, Addr_ID, Phone_No)
Employee(**Emp_ID**, Type, Salary_YR, Extension, Hire_Date, SSN_ID, Title)
Customer(**Cust_ID**, Acct_ID, SSN_ID)
Vehicle(**Car_ID**, Insurance_No)
Person(**SSN_ID**, F_Name, L_Name, M_Initial, B_Date, Sex, Race, Home_Phone, Work_Phone, Addr_ID)
Account(**Acct_ID**, Email, License_ID)
License(**License_ID**, License_No, License_State)
Insurance(**Insurance_No**, Vin_ID, Plate_No, Registered_State, Registered_Yr)
Vin(**Vin_ID**, Class, Features, Make, Color)
Car(**Car_ID**, Original_Price, Last_Oil_Change_Date, Last_Oil_Change_Mileage)
Address(**Addr_ID**, Street_Addr, City_Zip_ID, State_ID)
City_Zip(**City_Zip_ID**, City, Zipcode)
State(**State_ID**, State)
Hires(**Branch_ID**, Emp_ID)
Owns(**Branch_ID**, Car_ID)
Bought_From(**Cust_ID**, **Branch_ID**)
Transaction(**Emp_ID**, **Cust_ID**, **Car_ID**, **Trans_ID**)
Trans_Info(**Trans_ID**, Start_Date, End_Date, Describes, Miles_Used)



Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 3: Convert your ERD to Relational Database. (List your tables with the attributes). Identify your keys (primary and foreign).

Entity Tables

Branch(**Branch_ID**, Name, **Addr_ID**, Phone_No)

Name	**Null**	**Type**	**Key**
Branch_ID	Not Null	NUMBER(9),CHECK	Primary Key
Name	Null	VARCHAR2(12)	-
Addr_ID	Not Null	CHAR(30)	Foreign Key
Phone_No	Null	NUMBER(10)	-

Employee(**Emp_ID**, Type, Salary_YR, Extension, Hire_Date, **SSN_ID**, Title)

Name	**Null**	**Type**	**Key**
Emp_ID	Not Null	NUMBER(9)	Primary Key
Salary_YR	Null	NUMBER(5,2)	-
Extension	Null	NUMBER(3)	-
Hire_Date	Null	DATE	-
SSN_ID	Not Null	NUMBER(9)	Foreign Key
Title	Null	VARCHAR2(12)	-

Customer(**Cust_ID**, **Acct_ID, SSN_ID**)

Name	**Null**	**Type**	**Key**
Cust_ID	Not Null	NUMBER(9),CHECK	Primary Key
Acct_ID	Not Null	NUMBER(9)	Foreign Key
SSN_ID	Not Null	NUMBER(9)	Foreign Key

Vehicle(**Car_ID**, **Insurance_No**)

Name	**Null**	**Type**	**Key**
Car_ID	Not Null	NUMBER(9)	Primary Key
Insurance_No	Not Null	NUMBER(9)	Foreign Key

Person(**SSN_ID**, F_Name, L_Name, M_Initial, B_Date, Sex, Race, Home_Phone, Work_Phone, **Addr_ID**)

Name	**Null**	**Type**	**Key**
SSN_ID	Not Null	NUMBER(9)	Primary Key
F_Name	Null	VARCHAR2(12)	-
L_Name	Null	VARCHAR2(12)	-
M_Initial	Null	VARCHAR2(1)	-
B_Date	Null	DATE	-
Sex	Null	VARCHAR2(6)	-
Race	Null	VARCHAR2(12)	-

Home_Phone	Null	NUMBER(10)	-
Work_Phone	Null	NUMBER(10)	-
Addr_ID	Not Null	NUMBER(9)	Foreign Key

Account(**Acct_ID**, Email, **License_ID**)

Name	**Null**	**Type**	**Key**
Acct_ID	Not Null	NUMBER(9)	Primary Key
Email	Null	VARCHAR2(20)	-
License_ID	Not Null	NUMBER(9)	Foreign Key

License(**License_ID**, License_No, License_State)

Name	**Null**	**Type**	**Key**
License_ID	Not Null	NUMBER(9)	Primary Key
License_No	Null	NUMBER(9)	-
License_State	Null	VARCHAR2(15)	-

Insurance(**Insurance_No**, **Vin_ID**, Plate_No, Registered_State, Registered_YR)

Name	**Null**	**Type**	**Key**
Insurance_No	Not Null	NUMBER(9)	Primary Key
Vin_ID	Not Null	NUMBER(9)	Foreign Key
Plate_No	Null	VARCHAR2(7)	-
Registered_State	Null	VARCHAR2(15)	-
Registered_YR	Null	NUMBER(4)	-

Vin(**Vin_ID**, Class, Features, Make, Color)

Name	**Null**	**Type**	**Key**
Vin_ID	Not Null	NUMBER(9)	Primary Key
Class	Null	VARCHAR2(12)	-
Features	Null	VARCHAR2(12)	-
Make	Null	VARCHAR2(12)	-
Color	Null	VARCHAR2(12)	-

Car(**Car_ID**, Original_Price, Last_Oil_Change_Date, Last_Oil_Change_Mileage)

Name	**Null**	**Type**	**Key**
Car_ID	Not Null	NUMBER(9)	Primary Key
Original_Price	Null	NUMBER(5,2)	-
Last_Oil_Change_Date	Null	DATE	-
Last_Oil_Change_Mileage	Null	NUMBER(4)	-

Address(**Addr_ID**, Street_Addr, **City_Zip_ID**, **State_ID**)

Name	**Null**	**Type**	**Key**
Addr_ID	Not Null	NUMBER(9)	Primary Key
Street_Addr	Null	VARCHAR2(20)	-
City_Zip_ID	Not Null	NUMBER(6)	Foreign Key

State_ID	Not Null	NUMBER(15)	Foreign Key

City_Zip(**City_Zip_ID**, City, Zipcode)

Name	**Null**	**Type**	**Key**
City_Zip_ID	Not Null	NUMBER(6)	Primary Key
City	Null	VARCHAR2(15)	-
Zipcode	Null	NUMBER(6)	-

State(**State_ID**, State)

Name	**Null**	**Type**	**Key**
State_ID	Not Null	NUMBER(15)	Primary Key
State	Null	VARCHAR(15)	-

Relation Tables

Hires(**Branch_ID, Emp_ID**)

Name	**Null**	**Type**	**Key**
Branch_ID	Not Null	NUMBER(9)	Foreign Key
Emp_ID	Not Null	NUMBER(9)	Foreign Key

Owns(**Branch_ID, Car_ID**)

Name	**Null**	**Type**	**Key**
Branch_ID	Not Null	NUMBER(9)	Foreign Key
Car_ID	Not Null	NUMBER(9)	Foreign Key

Bought_From(**Cust_ID, Branch_ID**)

Name	**Null**	**Type**	**Key**
Cust_ID	Not Null	NUMBER(9)	Foreign Key
Branch_ID	Not Null	NUMBER(9)	Foreign Key

Transaction(**Emp_ID, Cust_ID, Car_ID, Trans_ID**)

Name	**Null**	**Type**	**Key**
Emp_ID	Not Null	NUMBER(9)	Foreign Key
Cust_ID	Not Null	NUMBER(9)	Foreign Key
Car_ID	Not Null	NUMBER(9)	Foreign Key
Trans_ID	Not Null	NUMBER(9)	Foreign Key

Trans_Info(**Trans_ID**, Start_Date, End_Date, Describes, Miles_Used)

Name	**Null**	**Type**	**Key**
Trans_ID	Not Null	NUMBER(9)	Primary Key
Start_Date	Null	DATE	-
End_Date	Null	DATE	-
Describes	Null	VARCHAR2(30)	-
Miles_Used	Null	NUMBER(4)	-

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 4: Normalize your tables step by step.

1st Normal Form

Employee(Emp_ID, F_Name, L_Name, M_Initial, Address, SSN, Type, Yearly_Salary, Phone_ext, DoB, Date_Hired, Sex, Race, Home_Phone)

Branch(Branch_ID, Name, Address, Phone_No, Number_of_Employees_in_that_Branch, Category, Manager_Name)

Vehicle(Car_ID, Registration, Class, Features, Make, Color, Last_Service, Vin_No, Original_Price, Insurance_No, Last_Oil_Change_Date, Last_Oil_Change_Mileage, Location_of_Last_Oil_Change)

Customer(SSN, ID, Name, Age, DoB, Home_Phone, Work_Phone, Street_Address, City, State, Zip_Code, Email, Acct_ID, Drivers_License_No, Drivers_License_State)

Transaction(Trans_ID, Start_Date, End_Date, Description, Price_Day, Tax, Insurance, Balance, Miles_Used_During_This_Transaction, Branch_Where_Car_Has_Been_Rented)

2nd Normal Form

Employee(Emp_ID, Type, Salary_YR, Ext, Date_Hired, SSN_ID, Phone_Ext)

P**erson**(F_Name, M_Initial, L_Name, Sex, SSN_ID, DoB, Race, Home_Phone)

Branch(Branch_ID, Branch_Name, Phone_No, Number_of_employees, Category, Manager_Name)

Address(State, City, Street, Zipcode, Country, ADDR_ID)

Vehicle(Last_Service, Original_Price, Insurance_No, Last_Oil_Change, Last_Oil_Change_Mileage, Location_of_Last_Oil_Change)

Registration(Vin_No, Color, Features, Class, Car_ID)

Customer(SSN_ID, Cust_ID, ADDR_ID, Email, Acct_ID, License_No, License_State)

Transaction(Trans_ID, Start_Date, End_Date, Description, Price_Day, Tax, Insurance, Balance, Miles_Used_During_This_Transaction, Branch_Where_Car_Has_Been_Rented)

3rd Normal Form

Branch(**Branch_ID**, Name, Addr_ID, Phone_No)

Employee(**Emp_ID**, Type, Salary_YR, Extension, Hire_Date, SSN_ID, Title)

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Customer(**Cust_ID**, Acct_ID, SSN_ID)

Vehicle(**Car_ID**, Insurance_No)

Person(**SSN_ID**, F_Name, L_Name, M_Initial, B_Date, Sex, Race, Home_Phone, Work_Phone, Addr_ID)

Account(**Acct_ID**, Email, License_ID)

License(**License_ID**, License_No, License_State)

Insurance(**Insurance_No**, Vin_ID, Plate_No, Registered_State, Registered_Yr)

Vin(**Vin_ID**, Class, Features, Make, Color)

Car(**Car_ID**, Original_Price, Last_Oil_Change_Date, Last_Oil_Change_Mileage)

Address(**Addr_ID**, Street_Addr, City_Zip_ID, State_ID)

City_Zip(**City_Zip_ID**, City, Zipcode)

State(**State_ID**, State)

Hires(**Branch_ID**, Emp_ID)

Owns(**Branch_ID**, Car_ID)

Bought_From(**Cust_ID**, **Branch_ID**)

Transaction(**Emp_ID**, **Cust_ID**, **Car_ID**, **Trans_ID**)

Trans_Info(**Trans_ID**, Start_Date, End_Date, Describes, Miles_Used)

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 5: Built your Oracle Tables with complete sets of constraints. Display using:

```
SELECT      Table_Name, Table_Type
FROM        CAT;
DESC        Table_Name;

SELECT      Owner, Constraint_Name, Constraint_Type, Table_Name,
            Search_Condition
FROM        User Constraints;

SELECT      *
FROM        User_Cons_Columns;

CREATE TABLE City_Zip
(City_Zip_ID        NUMBER(6)       NOT NULL,
City                VARCHAR2(15),
Zipcode             NUMBER(6),
PRIMARY KEY         (City_Zip_ID));

CREATE TABLE State
(State_ID           NUMBER(15)      NOT NULL,
State               VARCHAR2(15),
PRIMARY KEY         (State_ID));

CREATE TABLE Address
(Addr_ID            NUMBER(9)       NOT NULL,
Street_Addr         VARCHAR2(20),
City_Zip_ID         NUMBER(6)       REFERENCES City_Zip(City_Zip_ID),
State_ID            NUMBER(15)      REFERENCES State (State_ID),
PRIMARY KEY (Addr_ID));

CREATE TABLE Branch
(Branch_ID          NUMBER(9)   CHECK (Branch_ID Between 1000 AND 99999),
Name                VARCHAR2(12),
Addr_ID             NUMBER(9)       REFERENCES Address (Addr_ID),
Phone_No            NUMBER(10),
PRIMARY KEY         (Branch_ID));

CREATE TABLE Person
(SSN_ID             NUMBER(9)       NOT NULL,
F_Name              VARCHAR2(12),
L_Name              VARCHAR2(12),
M_Initial           VARCHAR2(1),
B_Date              DATE,
```

```
Sex             VARCHAR2(6),
Race            VARCHAR2(12),
Home_Phone      NUMBER(10),
Work_Phone      NUMBER(10),
Addr_ID         NUMBER(9)       REFERENCES Address (Addr_ID),
Primary Key     (SSN_ID));

CREATE TABLE Employee
(Emp_ID         NUMBER(9)       NOT NULL,
Salary_YR       NUMBER(5,2),
Extension       NUMBER(3),
Hire_Date       DATE,
SSN_ID          NUMBER(9)       REFERENCES Person (SSN_ID),
Title           VARCHAR2(12),
PRIMARY KEY     (Emp_ID));

CREATE TABLE License
(License_ID     NUMBER(9)       NOT NULL,
License_No      NUMBER(9),
License_State   VARCHAR2(15),
PRIMARY KEY     (License_ID));

CREATE TABLE Account
(Acct_ID        NUMBER(9)       NOT NULL,
Email           VARCHAR2(20),
License_ID      NUMBER(9)       REFERENCES License (License_ID),
PRIMARY KEY     (Acct_ID));

CREATE TABLE Customer
(Cust_ID        NUMBER(9) CHECK (Cust_ID Between 10000 AND 999999),
Acct_ID         NUMBER(9)       REFERENCES Account (Acct_ID),
SSN_ID          NUMBER(9)       REFERENCES Person (SSN_ID),
PRIMARY KEY     (Cust_ID));

CREATE TABLE Vin
(Vin_ID         NUMBER(9)       NOT NULL,
Class           VARCHAR2(12),
Features        VARCHAR2(12),
Make            VARCHAR2(12),
Color           VARCHAR2(12),
PRIMARY KEY     (Vin_ID));

CREATE TABLE Insurance
(Insurance_No   NUMBER(9)       NOT NULL,
Vin_ID          NUMBER(9)       REFERENCES Vin (Vin_ID),
```

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

```
Plate_No            VARCHAR2(7),
Registered_State    VARCHAR2(15),
Registered_YR       NUMBER(4),
PRIMARY KEY         (Insurance_No));

CREATE TABLE Vehicle
(Car_ID             NUMBER(9)       NOT NULL,
Insurance_No        NUMBER(9)       REFERENCES Insurance (Insurance_No),
PRIMARY KEY         (Car_ID));

CREATE TABLE Car
(Car_ID                     NUMBER(9)       NOT NULL,
Original_Price              NUMBER(5,2),
Last_Oil_Change_Date        DATE,
Last_Oil_Change_Mileage     NUMBER(4),
PRIMARY KEY                 (Car_ID));

CREATE TABLE Hires
(Branch_ID          NUMBER(9)       REFERENCES Branch (Branch_ID),
Emp_ID              NUMBER(9)       REFERENCES Employee (Emp_ID));

CREATE TABLE Owns
(Branch_ID          NUMBER(9)       REFERENCES Branch (Branch_ID),
Car_ID              NUMBER(9)       REFERENCES Car (Car_ID));

CREATE TABLE Bought_From
(Cust_ID            NUMBER(9)       REFERENCES Customer (Cust_ID),
Branch_ID           NUMBER(9)       REFERENCES Branch (Branch_ID));

CREATE TABLE Trans_Info
(Trans_ID           NUMBER(9)       NOT NULL,
Start_Date          DATE,
End_Date            DATE,
Describes           VARCHAR2(30),
Miles_Used          NUMBER(4),
PRIMARY KEY         (Trans_ID));

CREATE TABLE Transaction
(Emp_ID             NUMBER(9)       REFERENCES Employee (Emp_ID),
Cust_ID             NUMBER(9)       REFERENCES Customer (Cust_ID),
Car_ID              NUMBER(9)       REFERENCES Car (Car_ID),
Trans_ID            NUMBER(9)       REFERENCES Trans_Info (Trans_ID));
```

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 6: Insert minimum of three matching entries to each table. List the data for each table ((Display your data). Do not show your insert queries).

Select *
From City_Zip;

CITY_ZIP_ID	CITY	ZIPCODE
1	Cordova	21625
2	Easton	21601
3	Medway	2053
4	Arcadia	34265
5	San Diego	92127
6	Hogwarts	1
7	Narnia	2
8	The Shire	3
9	Westeros	4
10	Asgard	5

Select *
From State;

STATE_ID	STATE
1	Maryland
2	Massachusetts
3	Florida
4	California
5	Ohio
6	Texas
7	Hawaii
8	New York
9	Maine

Select *
From Address;

ADDR_ID	STREET_ADDR	CITY_ZIP_ID	STATE_ID
31864	Bittorf Ln	1	1
107	Park Ln	2	1
38	Maple St	3	2
8451	SW Riviera Dr	4	3
500	Sea World Dr	5	4
1	Gryffindor Tower	6	5
2	Castle Rd	7	6
3	Bilbos Hole	8	7
4	Winterfell	9	8
5	Throne Room In	10	9
6	Pacific Rd	6	1
7	Merry Rd	1	2
8	Sprig St	3	4
9	Cojak Ln	5	6
10	Memory Ln	7	8

Select *
From Branch;

BRANCH_ID	NAME	ADDR_ID	PHONE_NO
1000	Aylward Cars	31864	4103645753
1001	Kiln Cars	107	4108221453
1002	Toms Cars	38	3016870210
1003	Annes Cars	8451	3016871000
1004	Moures Cars	500	4108675409

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Select *
From Person;

SSN_ID	F_NAME	L_NAME	M_I	B_DATE	SEX	RACE	HOME_PHONE	WORK_PHONE	ADDR_ID
1	Harry	Potter	J	31-OCT-92	Male	White	1	1000000011	1
2	James	Penwell	T	15-NOV-87	Male	White	2	22	2
3	Bilbo	Baggins	F	01-JAN-92	Male	White	3	33	3
4	Aryan	Stark	X	13-APR-01	Female	European	4	44	4
5	Thor	Son	O	14-FEB-87	Male	White	5	55	5
6	John	Green	T	03-JUL-77	Male	White	6	66	6
8	Paul	McCartney	X	15-OCT-74	Male	White	8	88	8
9	George	Harrison	Y	04-NOV-77	Male	White	9	99	9
10	Ringo	Starr	Z	09-JUL-73	Male	White	10	10	10

Select *
From Employee;

EMP_ID	SALARY_YR	EXTENSION	HIRE_DATE	SSN_ID	TITLE
1	140	1	12-MAR-12	6	Manager
3	130	3	02-MAY-12	8	Manager
4	340	4	14-JUN-12	9	Manager
5	400	5	01-JUL-12	10	Manager

Select *
From License;

LICENSE_ID	LICENSE_NO	LICENSE_STATE
1	1	Massachusetts
2	2	Florida
3	3	Maryland
4	4	Hawaii
5	5	Texas

Select *
From Account;

ACCT_ID	EMAIL	LICENSE_ID
100000000	qwerty@gmail.com	1
200000000	uiopas@gmail.com	2
300000000	dfghjk@gmail.com	3
400000000	lzxcvb@gmail.com	4
500000000	nmqw@gmail.com	5

Select *
From Customer;

CUST_ID	ACCT_ID	SSN_ID
10000	100000000	1
10001	200000000	2
10002	300000000	3
10003	400000000	4
10004	500000000	5

Select *
From Vin;

VIN_ID	CLASS	FEATURES	MAKE	COLOR
1	Compact	4 Doors	Chevy	Blue
2	Economy	2 Doors	Pontiac	Red
3	Luxury	4 Doors	Ford	Green
4	Pickup	2 Doors	Toyota	Yellow
5	Van	4 Doors	Honda	Orange

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Select *
From Insurance;

INSURANCE_NO	VIN_ID	PLATE_NO	REGISTERED_STATE	REGISTERED_YR
1	1	aaaaaaa	Massachusetts	2001
2	2	bbbbbbb	Florida	2012
3	3	ccccccc	Maryland	1993
4	4	ddddddd	Hawaii	2000
5	5	eeeeeee	Texas	1999

Select *
From Vehicle;

CAR_ID	INSURANCE_NO
1	1
2	2
3	3
4	4
5	5

Select *
From Car;

CAR_ID	ORIGINAL_PRICE	LAST_OIL_CHANGE_DA	LAST_OIL_CHANGE_MILEAGE
1	340	01-DEC-14	2500
2	660	02-JAN-13	100
3	370	03-FEB-12	200
4	990	04-MAR-11	9800
5	830	05-APR-10	1110

Select *
From Hires;

BRANCH_ID	EMP_ID
1000	1
1002	3
1003	4
1004	5

Select *
From Owns;

BRANCH_ID	CAR_ID
1000	1
1001	2
1002	3
1003	4
1004	5

Select *
From Bought_From;

CUST_ID	BRANCH_ID
10000	1000
10001	1001
10002	1002
10003	1003
10004	1004

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014
Select *
From Trans_Info;

TRANS_ID	START_DATE	END_DATE	DESCRIBES	MILES_USED
1	17-MAY-13	18-MAY-13	Good Ride	9000
2	01-JUN-13	03-JUN-13	New Model	7000
3	27-JUL-13	30-JUL-13	Check tires	200
4	24-AUG-14	29-AUG-14	Engine Light	4500
5	01-OCT-14	03-OCT-14	Warp Speed	4000

Select *
From Transaction;

EMP_ID	CUST_ID	CAR_ID	TRANS_ID
1	10000	1	1
3	10002	3	3
4	10003	4	4
5	10004	5	5

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 7: Write the following queries and display the data:

1. Name, branch id, address of each branch

```
SELECT    Name, Branch_ID, Street_Addr, City, State
FROM      Branch, City_Zip, State, Address
WHERE     Branch.Addr_ID      =     Address.Addr_ID      AND
          Address.City_Zip_ID =     City_Zip.City_Zip_ID AND
          Address.State_ID    =     State.State_ID;
```

NAME	BRANCH_ID	STREET_ADDR	CITY	STATE
Aylward Cars	1000	Bittorf Ln	Cordova	Maryland
Killn Cars	1001	Park Ln	Easton	Maryland
Toms Cars	1002	Maple St	Medway	Massachusetts
Annes Cars	1003	SW Riviera Dr	Arcadia	Florida
Moures Cars	1004	Sea World Dr	San Diego	California

2. Name (first middle and last name), id, address, ssn, salary and type of employees

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Emp_ID, Street_Addr, City,
          State, Salary_YR, Title, Person.SSN_ID
FROM      Person, Employee, Address, City_Zip, State
WHERE     Person.SSN_ID       =     Employee.SSN_ID      AND
          Address.City_Zip_ID =     City_Zip.City_Zip_ID AND
          Address.State_ID    =     State.State_ID;
```

NAME	EMP_ID	STREET_ADDR	CITY	STATE	SALARY_YR	TITLE	SSN_ID
John T Green	1	Pacific Rd	Hogwarts	Maryland	140	Manager	6
Paul X McCartney	3	Sprig St	Medway	California	130	Manager	8
George Y Harrison	4	Cojak Ln	San Diego	Texas	340	Manager	9
Ringo Z Starr	5	Memory Ln	Narnia	New York	400	Manager	10

3. Name and age of customers

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Customers,
          TRUNC(MONTHS_BETWEEN(SYSDATE, B_Date) / 12 * (-1)) Age
FROM      Person, Customer
WHERE     Person.SSN_ID       =     Customer.SSN_ID;
```

CUSTOMER	AGE
Harry J Potter	77
James T Penwell	72
Bilbo F Baggins	77
Aryan X Stark	-13
Thor O Son	72

4. Transaction number, starting date, and total mile of each transaction

```
SELECT    Transaction.Trans_ID, Start_Date, Miles_Used
FROM      Transaction, Trans_Info
WHERE     Transaction.Trans_ID      =     Trans_Info.Trans_ID;
```

TRANS_ID	START_DATE	MILES_USED
1	17-MAY-13	9000
3	27-JUL-13	200
4	24-AUG-14	4500
5	01-OCT-14	4000

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

5. List of car id, make, and color of each vehicle

```
SELECT    Car_ID, Make, Color
FROM      Vin;
```

CAR_ID	MAKE	COLOR
1	Chevy	Blue
2	Pontiac	Red
3	Ford	Green
4	Toyota	Yellow
5	Honda	Orange

6. Name, branch id, phone number of branches that have less than 5 employees

```
SELECT    Branch.Name, Branch.Branch_ID, Branch.Phone_No
FROM      Branch, Employee, Hires
WHERE     Hires.Emp_ID      =     Employee.Emp_ID     AND
          Hires.Branch_ID   =     Branch.Branch_ID    AND
          Hires.Branch_ID   <     Hires.Emp_ID('5');
```

NAME	BRANCH_ID	PHONE_NO
Aylward Cars	1000	4103645753
Toms Cars	1002	3016870210
Annes Cars	1003	3016871000
Moures Cars	1004	4108675409

7. Transaction id, price per day and drivers' license state of transactions for customers who brought additional insurance

```
SELECT    Transaction.Trans_ID, Car.Original_Price, License.License_State
FROM      Transaction, Car, License, Insurance, Customer
WHERE     Transaction.Car_ID   =     Car.Car_ID          AND
          Transaction.Cust_ID  =     Customer.Acct_ID;
```

NO ROWS SELECTED

8. Car id, VIN numbers of vehicles that have done the oil change within last two months

```
SELECT    Car.Car_ID, Vin_ID, Car.Last_Oil_Change_Date
FROM      Car, Vehicle, Insurance
WHERE     Car.Car_ID                 =     Vehicle.Car_ID        AND
          Vehicle.Insurance_No       =     Insurance.Vin_ID      AND
          Car.Last_Oil_Change_Date   >     ADD_MONTHS(SYSDATE, -2);
```

CAR_ID	VIN_ID	LAST_OIL_CHANGE_DA
1	1	01-DEC-14

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

9. Branch name, number of employee, and name of the manager in each branch

```
SELECT    Branch.Name, Employee.Title, F_Name||' '||M_Initial||' '||L_Name
          Name, Employee.Emp_ID Employees
FROM      Branch, Employee, Hires, Person
WHERE     Hires.Branch_ID   =   Branch.Branch_ID   AND
          Hires.Emp_ID      =   Employee.Emp_ID    AND
          Person.SSN_ID     =   Employee.SSN_ID;
```

NAME	TITLE	NAME	EMPLOYEES
Aylward Cars	Manager	John T Green	1
Toms Cars	Manager	Paul X McCartney	3
Annes Cars	Manager	George Y Harrison	4
Moures Cars	Manager	Ringo Z Starr	5

10. Total number of cars

```
SELECT    COUNT(Car_ID) Cars
FROM      Car;
```

CARS
5

11. List of employees who are also customers

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name
FROM      Person, Employee, Customer
WHERE     Person.SSN_ID     =   Employee.SSN_ID    AND
          Person.SSN_ID     =   Customer.SSN_ID;
```

12. Sort the name of customers alphabetic. Display the name and phone area code

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name,
          LPAD(Person.Home_Phone, 3)
FROM      Person, Customer
WHERE     Person.SSN_ID     =   Customer.SSN_ID
ORDER BY  L_Name;
```

NAME	LPAD(PERS
Bilbo F Baggins	3
James T Penwell	2
Harry J Potter	1
Thor O Son	5
Aryan X Stark	4

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

13. Sort each branch by the category. List name of employees who work at that branch

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Branch.Name Branch
FROM      Person, Hires, Employee, Branch
WHERE     Hires.Emp_ID      =     Employee.Emp_ID    AND
          Employee.SSN_ID   =     Person.SSN_ID      AND
          Branch.Branch_ID  =     Hires.Branch_ID;
```

NAME	BRANCH
John T Green	Aylward Cars
Paul X McCartney	Toms Cars
George Y Harrison	Annes Cars
Ringo Z Starr	Moures Cars

14. List the name of employee and birth date who has the highest salary

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Person.B_Date,
          Employee.Salary_YR
FROM      Person, Hires, Employee, Branch
WHERE     Hires.Emp_ID      =     Employee.Emp_ID    AND
          Employee.SSN_ID   =     Person.SSN_ID      AND
          Branch.Branch_ID  =     Hires.Branch_ID
ORDER BY  Salary_YR         DESC;
```

NAME	B_DATE	SALARY_YR
Ringo Z Starr	09-JUL-73	400
George Y Harrison	04-NOV-77	340
John T Green	03-JUL-77	140
Paul X McCartney	15-OCT-74	130

15. Name the employee and birth date who has the highest salary

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Person.B_Date,
          Employee.Salary_YR
FROM      Person, Hires, Employee, Branch
WHERE     Hires.Emp_ID      =     Employee.Emp_ID    AND
          Employee.SSN_ID   =     Person.SSN_ID      AND
          Branch.Branch_ID  =     Hires.Branch_ID
ORDER BY  Salary_YR         DESC
LIMIT     1;
```

NAME	B_DATE	SALARY_YR
Ringo Z Starr	09-JUL-73	400

16. List the name of employee whose first name start with an 's' and end with a 'd'

```
SELECT    F_Name||' '||M_Initial||' '||L_Name
FROM      Person
WHERE     F_Name    LIKE    'S%'    AND
          F_Name    LIKE    '%d';
```

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

17. Customer names sorted by city, state, and the color of the vehicles they rented

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Address.Street_Addr,
          City_Zip.City, State.State, Vin.Color
FROM      Person, Customer, Address, City_Zip, State, Vin, Insurance, Vehicle,
          Transaction
WHERE     Person.SSN_ID          =    Customer.SSN_ID       AND
          Person.Addr_ID         =    Address.Addr_ID       AND
          Address.City_Zip_ID    =    City_Zip.City_Zip_ID  AND
          Address.State_ID       =    State.State_ID        AND
          Insurance.Vin_ID       =    Vin.Vin_ID            AND
          Vehicle.Car_ID         =    Transaction.Car_ID    AND
          Customer.Cust_ID       =    Transaction.Cust_ID   AND
          Person.SSN_ID          =    Customer.SSN_ID;
```

NAME	STREET_ADDR	CITY	STATE	COLOR
Harry J Potter	Gryffindor Tower	Hogwarts	Ohio	Blue
Harry J Potter	Gryffindor Tower	Hogwarts	Ohio	Red
Harry J Potter	Gryffindor Tower	Hogwarts	Ohio	Green
Harry J Potter	Gryffindor Tower	Hogwarts	Ohio	Yellow
Harry J Potter	Gryffindor Tower	Hogwarts	Ohio	Orange
Bilbo F Baggins	Bilbos Hole	The Shire	Hawaii	Blue
Bilbo F Baggins	Bilbos Hole	The Shire	Hawaii	Red
Bilbo F Baggins	Bilbos Hole	The Shire	Hawaii	Green
Bilbo F Baggins	Bilbos Hole	The Shire	Hawaii	Yellow
Bilbo F Baggins	Bilbos Hole	The Shire	Hawaii	Orange
Aryan X Stark	Winterfell	Westeros	New York	Blue
Aryan X Stark	Winterfell	Westeros	New York	Red
Aryan X Stark	Winterfell	Westeros	New York	Green
Aryan X Stark	Winterfell	Westeros	New York	Yellow
Aryan X Stark	Winterfell	Westeros	New York	Orange
Thor O Son	Throne Room In	Asgard	Maine	Blue
Thor O Son	Throne Room In	Asgard	Maine	Red
Thor O Son	Throne Room In	Asgard	Maine	Green
Thor O Son	Throne Room In	Asgard	Maine	Yellow
Thor O Son	Throne Room In	Asgard	Maine	Orange

18. Name of customer, total mileage, and number of cars they rented

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Trans_Info.Miles_Used
FROM      Person, Customer, Trans_Info, Transaction
WHERE     Person.SSN_ID          =    Customer.SSN_ID       AND
          Trans_Info.Trans_ID    =    Transaction.Trans_ID  AND
          Transaction.Cust_ID    =    Customer.Cust_ID;
```

NAME	MILES_USED
Harry J Potter	9000
Bilbo F Baggins	200
Aryan X Stark	4500
Thor O Son	4000

19. Today's date, customer name, customer's birth date, and how many days are left to the customer's birth date. For customers with birth date in the present month. (extract month from SYSDATE)

```
SELECT    SYSDATE, F_Name||' '||M_Initial||' '||L_Name, B_Date,
          (SYSDATE – B_Date - 365 * (-1)
FROM      Person, Customer
WHERE     Person.SSN_ID          =    Customer.SSN_ID       AND
          Person.B_Date          =    ADD_MONTHS(SYSDATE, 0);
```

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

20. Name of employees, Home phone number, the year, the month, and the day they born; sorted from the oldest to the youngest

```
SELECT      F_Name||' '||M_Initial||' '||L_Name Name, Home_Phone,
            To_CHAR(B_Date, 'YYYY-MON-DD')
FROM        Person, Employee
WHERE       Person.SSN_ID       =       Employee.SSN_ID
ORDER BY    B_Date DESC;
```

NAME	HOME_PHONE	TO_CHAR(B_DATE,'YYYY-MON-DD')
George Y Harrison	9	2077-NOV-04
John T Green	6	2077-JUL-03
Paul X McCartney	8	2074-OCT-15
Ringo Z Starr	10	2073-JUL-09

21. Vin number, car id, original price, total mileage on that car (add the rented mileage)

```
SELECT      Car.Car_ID, Original_Price, Trans_Info.Miles_Used +
            Last_Oil_Change_Mileage Mileage, Vin_ID
FROM        Car, Trans_Info, Transaction, Insurance
WHERE       Transaction.Trans_ID    =       Car.Car_ID          AND
            Transaction.Trans_ID    =       Trans_Info.Trans_ID AND
            Vin_ID                  =       Transaction.Trans_ID;
```

CAR_ID	ORIGINAL_PRICE	TRANS_INFO.MILES_USED+LAST_OIL_CHANGE_MILEAGE	VIN_ID
1	340	11500	1
3	370	400	3
4	990	14300	4
5	830	5110	5

22. List of customer without email address

```
SELECT      F_Name||' '||M_Initial||' '||L_Name
FROM        Person, Customer, Account
WHERE       Person.SSN_ID       =       Customer.SSN_ID     AND
            Customer.Acct_ID    =       Account.Acct_ID     AND
            Account.Email       IS NULL;
```

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

Chapter 8: List 15 queries:

10 queries for single row functions

1. Convert all Employee First NAMES to UPPER CASE and all Employee Last NAMES to LOWER CASE then Display the Query:

```
SELECT      UPPER( F_Name) First, LOWER(L_Name) Last
FROM        Person, Employee
WHERE       Person.SSN_ID     =     Employee.SSN_ID;
```

FIRST	LAST
JOHN	green
PAUL	mccartney
GEORGE	harrison
RINGO	starr

2. Convert all Customers First NAMES to LOWER CASE and all Customers Last NAMES to UPPER CASE, then Display the Query.

```
SELECT      LOWER( F_Name) First, UPPER(L_Name) Last
FROM        Person, Customer
WHERE       Person.SSN_ID     =     Customer.SSN_ID;
```

FIRST	LAST
harry	POTTER
james	PENWELL
bilbo	BAGGINS
aryan	STARK
thor	SON

3. Show the NAME of Employee that USE EMAIL provider Yahoo, G-Mail:

```
SELECT      Account.Email, Account.Acct_ID, F_Name||' '||M_Initial||' '||L_Name
            Name
FROM        Account , Person
WHERE       Account.Email LIKE 'gmail';
```

4. Display all BRANCH NAMES in Alphabetical order:

```
SELECT      Branch.Name
FROM        Branch
ORDER BY    Branch.Name ASC;
```

NAME
Annes Cars
Aylward Cars
Kilin Cars
Moures Cars
Toms Cars

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

5. Display a list of all CUSTOMERS Birth dates in the format YY-MM-DD, In order of oldest to youngest:

```
SELECT      F_Name||' '||M_Initial||' '||L_Name Name,
            To_CHAR(B_Date, 'YYYY-MON-DD')
FROM        Person, Customer
WHERE       Person.SSN_ID = Customer.SSN_ID
ORDER By    B_Date;
```

NAME	TO_CHAR(B_DATE,'YYYY-MON-DD')
Aryan X Stark	2001-APR-13
Thor O Son	2087-FEB-14
James T Penwell	2087-NOV-15
Bilbo F Baggins	2092-JAN-01
Harry J Potter	2092-OCT-31

6. Display a list of all EMPLOYEES are paid less yearly than the employee with highest yearly salary and by how much:

```
SELECT      F_Name||' '||L_Name, a.Salary_YR - b.Salary_YR DIFFERENCE
FROM        Person, Employee a, employee b
WHERE       a.Salary_YR > b.Salary_YR;
```

F_NAME||''||L_NAME	DIFFERENCE
Harry Potter	60
James Penwell	60
Bilbo Baggins	60
Aryan Stark	60
Thor Son	60
John Green	60

7. Display a list of all EMPLOYEE, that lives on a lane (Example 231 Wood **Ln**. Cumberland MD 21532):

```
SELECT      F_Name||' '||M_Initial||' '||L_Name name, Address.Street_Addr
FROM        Person, Employee, Address
WHERE       Person.SSN_ID       =       Employee.SSN_ID         AND
            Person.Addr_ID      =       Address.Addr_ID         AND
            Address.Street_Addr         LIKE                    '%Ln';
```

NAME	STREET_ADDR
George Y Harrison	Cojak Ln
Ringo Z Starr	Memory Ln

8. Display a list of all EMPLOYEE that currently live in the STATE of Maryland:

```
SELECT      F_Name||' '||M_Initial||' '||L_Name Name, State.State
FROM        Person, State, Address, Employee
WHERE       Person.Addr_ID      =       Address.Addr_ID     AND
            Address.State_ID    =       State.State_ID      AND
            Person.ssn_ID       =       Employee.SSN_ID     AND
            State.State         Like    'Maryland';
```

NAME	STATE
John T Green	Maryland

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

9. Display a list of all CARS that have '4' Doors, with the COLOR and MODEL of CAR:

```
SELECT    Class, Features, Make, Color
FROM      Vin
WHERE     Vin.Features    LIKE    '4%';
```

CLASS	FEATURES	MAKE	COLOR
Compact	4 Doors	Chevy	Blue
Luxury	4 Doors	Ford	Green
Van	4 Doors	Honda	Orange

10. Display a list showing the ZIPCODE of EMPLOYEE who have also rented a car (also a customer) in a DECENDING order (by last name):

```
SELECT    City_Zip.Zipcode
FROM      Person, Customer, City_Zip, Employee
WHERE     Person.SSn_ID     =     Customer.SSn_ID     AND
          Person.SSN_ID     =     Employee.SSN_ID
ORDER BY  City_Zip.zipcode  DESC;
```

5 queries for multiple row functions

1. Display the CUSTOMER Born in October after the year 1970 with there full name:

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name
FROM      Person, Customer
WHERE     Person.SSN_ID                =     Customer.SSN_ID     AND
          TO_CHAR(B_Date, 'MON')       =     'OCT'               AND
          TO_CHAR(B_Date, 'YYYY')      >     '1970';
```

NAME
Harry J Potter

2. Display the EMPLOYEE with yearly salary greater than 200 and was born in the month of October:

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name
FROM      Person, Employee
WHERE     Person.SSN_ID                =     Employee.SSN_ID     AND
          TO_CHAR(B_Date,'MON')        =     'OCT'               AND
          (Salary_YR> '200');
```

3. Display the name of all CUSTOMER along with the sum of DAYS they rented their individual CARS:

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name,
          End_Date - Start_Date DIFFERENCE
FROM      Person, Customer, Trans_Info, Transaction
WHERE     Person.SSN_ID        =     Customer.SSN_ID          And
          Trans_Info.Trans_ID  =     Transaction.Trans_ID     And
          Transaction.Cust_ID  =     Customer.Cust_ID;
```

NAME	DIFFERENCE
Harry J Potter	1
Bilbo F Baggins	2
Aryan X Stark	4
Thor O Son	1

4. Display the AVERAGE pay of all EMPLOYEES that hold a MANAGERIAL position:

```
SELECT    AVG(Employee.Salary_YR)
FROM      Person, Employee
WHERE     Person.SSN_ID     =     Employee.SSN_ID     AND
          Employee.Title    =     'Manager';
```

AVG(EMPLOYEE.SALARY_YR)
252.5

5. Display the amount of PERSONS(employees + customers) that live in the state of Hawaii:

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, State.State
FROM      Person, State, Address
WHERE     Person.Addr_ID    =     Address.Addr_ID     AND
          Address.State_ID  =     State.State_ID      AND
          State.State       LIKE  'Hawaii';
```

NAME	STATE
Bilbo F Baggins	Hawaii

Chapter 9: List 10 queries:

5 inner join

1. Inner Join Employee and Customer to display a list of names of Employees that are also Customer:

```
SELECT    Employee.SSN_ID
FROM      Employee          INNER JOIN  Customer      ON
          Employee.SSN_ID   =           Customer.SSN_ID;
```

2. Inner Join Branch and ADDRESS_ID to display all Addresses_ID from each branch by name:

```
SELECT    Branch.Addr_ID, Name
FROM      Branch            INNER JOIN  Address       ON
          Branch.Addr_ID    =           Address.Addr_ID;
```

ADDR_ID	NAME
31864	Aylward Cars
107	Killn Cars
38	Toms Cars
8451	Annes Cars
500	Moures Cars

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

3. Inner Join Person and ADDRESS_ID to display all ADDRESS_ID for each Person by Name(First middle last):

```
SELECT    Person.Addr_ID, F_Name||' '||M_Initial||' '||L_Name
FROM      Person          INNER JOIN      Address     ON
          Person.Addr_ID  =               Address.Addr_ID;
```

ADDR_ID	F_NAME||''||M_INITIAL||''||L_NAME
1	Harry J Potter
2	James T Penwell
3	Bilbo F Baggins
4	Aryan X Stark
5	Thor O Son
6	John T Green
8	Paul X McCartney
9	George Y Harrison
10	Ringo Z Starr

4. Inner Join Person and Employee to display all Employees fromPerson, list Name(First middle last):

```
SELECT    Person.SSN_ID, F_Name||' '||M_Initial||' '||L_Name
FROM      Person         INNER JOIN      Employee    ON
          Person.SSN_ID  =               Employee.SSN_ID;
```

SSN_ID	F_NAME||''||M_INITIAL||''||L_NAME
6	John T Green
8	Paul X McCartney
9	George Y Harrison
10	Ringo Z Starr

5. Inner Join Account and License by License to the Accounts Email:

```
SELECT    Account.License_ID, Account.Email
FROM      Account            INNER JOIN      License     ON
          Account.License_ID =               License.License_ID;
```

LICENSE_ID	EMAIL
1	qwerty@gmail.com
2	uiopas@gmail.com
3	dfghjk@gmail.com
4	lzxcvb@gmail.com
5	nmqw@gmail.com

5 outer join

1. Using Right Outer Join, Join both the Employee and Person Tables by SSN_ID:

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name, Salary_YR
FROM      Employee        RIGHT OUTER JOIN  Person ON
          Person.SSN_ID   =                 Employee.Salary_YR;
```

A	SALARY_YR
Harry	
James	
Bilbo	
Aryan	
Thor	
John	
Paul	
George	
Ringo	

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

2. Using Right Outer Join to Join Customer to Person by SSN_ID:

```
SELECT    F_Name||' '||M_Initial||' '||L_Name Name , Cust_ID
FROM      Customer    RIGHT OUTER JOIN  Person    ON
          Person.SSN_ID    =            Customer.Cust_ID;
```

NAME	CUST_ID
Harry J Potter	
James T Penwell	
Bilbo F Baggins	
Aryan X Stark	
Thor O Son	
John T Green	
Paul X McCartney	
George Y Harrison	
Ringo Z Starr	

3. Using Right Outer Join to Join Account and Customer by Cust_ID:

```
SELECT    Account.Acct_ID Account, Cust_ID
FROM      Customer    RIGHT OUTER JOIN  Account   ON
          Account.Acct_ID  =            Customer.Cust_ID;
```

ACCOUNT	CUST_ID
100000000	
200000000	
300000000	
400000000	
500000000	

4. Using Right Outer Join to join Tables Vin by Vin_ID to Table Insurance by Insurance_No:

```
SELECT    Vin.Vin_ID, Insurance_No
FROM      Insurance   RIGHT OUTER JOIN  Vin       ON
          Vin.Vin_ID       =            Insurance.Insurance_No;
```

VIN_ID	INSURANCE_NO
1	1
2	2
3	3
4	4
5	5

5. Using Right Outer Join to join Tables Vehicle by Car_ID to Table Insurance by Insurance_No:

```
SELECT    Vehicle.Car_ID , Insurance.Insurance_No
FROM      Insurance   RIGHT OUTER JOIN  Vehicle   ON
          Vehicle.Car_ID   =            Insurance.Insurance_No;
```

CAR_ID	INSURANCE_NO
1	1
2	2
3	3
4	4
5	5

Chapter 10: List of 5 sub-queries.

1. Using Sub-Queries display a list of Employee Social Security Numbers from Entity Person:

```
SELECT    Employee.SSN_ID
FROM      Employee
```

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014

```
WHERE       SSN_ID      IN
            (SELECT     SSN_ID
            FROM        Person
            WHERE       Person.SSN_ID = Employee.SSN_ID);
```

SSN_ID
6
8
9
10

2. Using Sub-Queries display a list of Customer Social Security Numbers from the Entity Person:

```
SELECT      Customer.SSN_ID
FROM        Customer
WHERE       SSN_ID      IN
            (SELECT     SSN_ID
            FROM        Person
            WHERE       Person.SSN_ID = Customer.SSN_ID);
```

SSN_ID
1
2
3
4
5

3.

```
SELECT      Employee.SSN_ID, F_Name||' '||M_Initial||' '||L_Name Name
FROM        Employee, Person
WHERE       Employee.SSN_ID     =       Person.SSN_ID       AND
            Title       IN
            (SELECT     Title
            FROM        Employee
            WHERE       Employee.Title =                    'Manager');
```

SSN_ID	NAME
10	Ringo Z Starr
9	George Y Harrison
8	Paul X McCartney
6	John T Green

4.

```
SELECT      Vin.Vin_ID, Plate_No
FROM        Vin, Insurance
WHERE       Vin.Vin_ID      =       Insurance.Vin_ID      AND
            Insurance.Plate_No  IN
            (SELECT     Plate_No
            FROM        Insurance
            WHERE       Insurance.Plate_No  LIKE    '%a');
```

VIN_ID	PLATE_NO
1	aaaaaaa

Jacob Aylward – CS440F142
Database Project COSC 440
Nov. 18th 2014
5.

```
SELECT    Address.Addr_ID, Race
FROM      Address, Person
WHERE     Address.ADDR_ID = Person.Addr_ID AND
          Race IN
          (SELECT Race
          FROM Person
          WHERE Race Like 'White');
```

ADDR_ID	RACE
10	White
9	White
8	White
6	White
5	White
3	White
2	White
1	White

Chapter 11: 5 views involving more than one table.

1.

2.

3.

4.

5.