**Round # 1**

Draw the Entity Relationship Diagram (ERD) for the application described below. Specify the entities, the relationships between the entities, as well as the cardinality and optionality of the relationships. Clearly label all primary and foreign keys. Indicate what type of notation (crow’s foot, Chen, or other) you are using to show cardinality (one or many) and optionality (mandatory or optional) at the top of your ERD. You have to clearly specify primary keys and foreign keys as needed to simplify the database design, ***Save your ERD on \\server1\Challenge Week 2019\DataBase Designing\DB-<<Group code>>***

**Scenario:**

A database is to be designed for a Car Rental Co. (CRC). The information required includes a description of Cars, Subcontractors (i.e., garages), company expenditures, Company revenues and customers. Cars are to be described by such data as make, model, year of production, and engine size, and fuel type, number of passengers, registration number, purchase price, purchase date, rent price and insurance details. It is the company policy not to keep any car for a period exceeding one year. All major repairs and maintenance are done by Subcontractors (i.e., franchised garages), with whom CRC has long-term agreements. Therefore the data about garages to be kept in the database includes garage names, addresses, range of Services and the like. Some garages require payments immediately after a repair has been made, with others CRC has made arrangements for Credit facilities. Company expenditures are to be registered for all outgoings connected with purchases, repairs, maintenance, insurance etc. Similarly, the cash inflow coming from all Sources – Car hire, car sales, insurance claims – must be kept of file. CRC maintains reasonably stable client base. For this privileged category of customers special credit card facilities are provided. These customers may also book in advance a particular car. These reservations can be made for any period of time up to one month. Casual Customers must pay a deposit for an estimated time of rental, unless they wish to pay by Credit card. All major credit cards care accepted. Personal details (such as name, address, telephone number, driving license number) about each customer are kept in the database.

**Round 2: Queries**

*Save all queries in one SQL script named* ***Team<<group code>>\_All Queries***. Comment your code so that each of the queries are clearly labeled as ***Query1***, ***Query2***, etc. When you are done, upload your script to the contest server.

1. List reservation count according to cabin\_type. Save as **Query1.**
2. List the names of all cabins that have no current reservations. Order by the cabin name. Save as **Query2.**
3. Write a query that shows the packages that have 4 or more activities included with that package. List package name and the number of activities included with that package. Put in descending order by the number of activities. Save as **Query3.**
4. Create a query that lists all cabins that accommodate more than the average number of people for *its own cabin type.* Save as **Query4.**
5. This query will select information about each reservation. List the reservation number, the name of the package, and the total cost for every guest listed on that reservation. For example, if a reservation has 3 people, and it is for the “American Winter Season”, which has a price of $650 per person, the total cost would be 3 \* 650 = 1950. Format the data as shown in the results below. Only include those reservations where the total cost exceeds $2,000 and order by the reservation number. Save as **Query5**.
6. Create a view named ***vw\_PackageRevenue*** that will list the total revenue generated by each vacation package. Revenue is calculated as cost per person multiplied by the number of guests who have signed up for that package. Generate the list by retrieving data through the view.Save as **Query6**.
7. Create a stored procedure named ***sp\_CabinInventory*** that will accept a cabin type as a parameter and prepare a list of all cabins of the cabin type specified along with cabin name, number of guests it can accommodate, reservation number for each cabin, guest’s arrival date and number of guests on that reservation. Save the script to create the stored procedure as **Query7.**
8. List the guest’s name, arrival date and departure date (this is determined by the number of days in the package.) Only list those guests whose packages are 5, 6, or 7 days long. Save as **Query8.**
9. Create DDL triggers to stop creation, modification or deletion of the tables. Save as **query9.**
10. Create a trigger that logs inserted, updated and deleted records of guests. Save as **query10**

**Evaluation Strategy**

Total Marks: 100

Database Design: 30 Marks

Query Building: 70 Marks.