**Data Management Homework 1**

**This homework includes part of our in-class exercise. Please submit 3 items listed below:**

**1. Write a paragraph with more than 150 words explaining the importance of index and keys in RDBMS.**

Having an index in tables, makes data retrieval faster as the operation will not have to search every row in the database table every time the database table is accessed. Keys are used to speed up access to records in a database. Keys are used to establish and identify relationships between tables and also to uniquely identify any record or row of data inside a table. For instance primary keys are used when linking tables. Databases that have thousands to millions of records, make it harder for one to access specific records especially if there are duplicates present in the database, hence keys make it easy to identify specific records. Primary keys prevent null values from being maintained in a database. Indexes make sure data is stored in an orderly fashion. Foreign keys ensure database maintains referential integrity. This means the database monitors the data inserted into the parent and child tables.  For example; if the user attempts to enter a record in the Orders table using a CustomerID that is not found in the Customers table, the database will reject that entry and display a warning message. Another obvious advantage for using foreign key constraints is improved performance.  By including information on how tables are joined, SQL Server can easily determine how it’s going to retrieve data when using those joins. Foreign key assist database developers to determine how the database is designed. For example typically, when you use the ‘View Dependencies…’ option for either table, you will be greeted with a visible hierarchy showing how the Customers and Orders tables are related.

**2. Write a paragraph with more than 150 words explaining the importance of database design.**

A database design is well planned and structured outline used by a database to store and manage data. A design ensures data consistency whereby a database is designed to store only useful and required data. Tables in a design allows data to be consistent. Primary key implementation also ensures data consistency. Data uniformity is ensured through cascading, whereby records in child tables that are linked to valid parent records exist. In addition a good database design reduces data redundancy hence curbing unnecessary huge volume of data. Keys are used to refer to possible values of a table to avoid duplication. Thus when the value is altered only the change is reflected once in the primary table. A well designed database uses simpler queries and quick executions due to optimized relationships. Generally a good database relies on a good design. Thus a good design ensures that there is room for easy maintenance. A poorly designed database may cause damage to stored procedures, views and functions during the fixing of minor breakdowns.

**3. Using the following dataset (fake data), you will need to design a database with multiple tables. Each table should have its own primary key and the tables with relationship should be connected with foreign key. You can use MySQL workbench to make the design or draw your own plot using other tools. Write CREATE TABLE query (and ALTER TABLE if needed) to create these tables and insert the following data into your tables.**

|  |
| --- |
| **D\_DRIVER** |
| DriverLicence (PK) |
| DrierName |
| DriverAddress |

**ENTITY RELATIONSHIP DIAGRAM**

|  |
| --- |
| **F\_TRANSPORT** |
| TransportID (PK) |
| CarPlate (FK) |
| CarMaker |
| CarColor |
| DriverLicence (FK) |
| DriverName |
| DriverAddress |
| PolicyNumber (FK) |
| PolicyPeriod |
| InsuranceName |
| InsuranceAddress |

|  |
| --- |
| **D\_INSURANCE** |
| PolicyNumber (PK) |
| PolicyPeriod |
| InsuranceName |
| InsuranceAddress |

|  |
| --- |
| **D\_CAR** |
| CarPlate (PK) |
| CarMaker |
| CarColor |

**QUERY FOR TABLE CREATION**

**D\_CAR TABLE**

CREATE TABLE Schema1.D\_CAR (

CarPlate VARCHAR(50) NOT NULL,

CarMaker VARCHAR(50),

CarColor VARCHAR(50),

CONSTRAINT PK\_CAR PRIMARY KEY (CarPlate)

);

**DRIVER TABLE**

CREATE TABLE D\_DRIVER (

DriverLicence VARCHAR (50),

DriverName VARCHAR (50),

DriverAddress VARCHAR (50),

InsuranceAddress VARCHAR (50),

CONSTRAINT PK\_DRIVER PRIMARY KEY (DriverLicence)

);

**INSURANCE TABLE**

CREATE TABLE D\_INSURANCE (

PolicyNumber, VARCHAR (50),

Policy Period VARCHAR (50),

InsuranceName VARCHAR (50),

InsuranceAddress VARCHAR (50),

CONSTRAINT PK\_INSURANCE PRIMARY KEY (PolicyNumber)

);

**TRANSPORT TABLE**

CREATE TABLE F\_TRANSPORT (

TransportID VARCHAR (50),

CarPlate VARCHAR (50),

CarMaker VARCHAR (50),

CarColor VARCHAR (50),

DriverLicence VARCHAR (50)

DriverName VARCHAR (50),

DriverAddress VARCHAR (50),

PolicyNumber, VARCHAR (50),

Policy Period VARCHAR (50),

InsuranceName VARCHAR (50),

InsuranceAddress VARCHAR (50),

Constraint PK\_TRANSPORT PRIMARY\_KEY (TransportID),

constraint fk\_Car foreign key(CarPlate) references D\_CAR(CarPlate),

constraint fk\_Driver foreign key(DriverLicence) references D\_DRIVER(DriverLicence),

constraint fk\_Insurance foreign key(PolicyNumber) references D\_INSURANCE(PolicyNumber)

);