Spring- 2017

1.a)

A collection of interrelated data and a set of programs to access these data. Its primary goal is to provide a way store and retrieve database info that is both convenient and efficient.

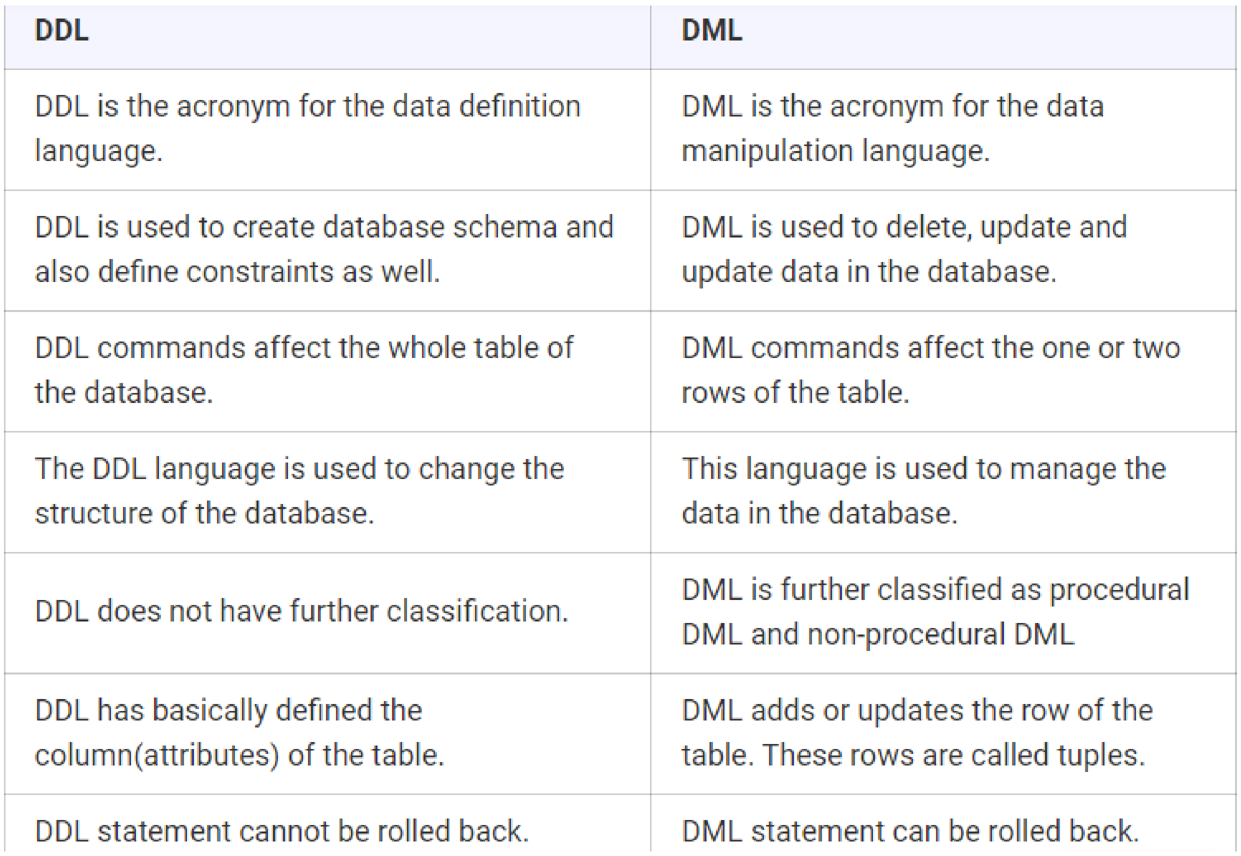
Examples:

1) Enterprise info
2) Banking and Finance
3) Universities
4) Airlines
5) Telecommunication

1.b)

Data model is collection of conceptual tools for describing data, data relationships, data semantics and consistency constraints. It provides a way to describe the design of a database at all levels.

DDL vs DML



DDL	DML
DDL is the acronym for the data definition language.	DML is the acronym for the data manipulation language.
DDL is used to create database schema and also define constraints as well.	DML is used to delete, update and update data in the database.
DDL commands affect the whole table of the database.	DML commands affect the one or two rows of the table.
The DDL language is used to change the structure of the database.	This language is used to manage the data in the database.
DDL does not have further classification.	DML is further classified as procedural DML and non-procedural DML
DDL has basically defined the column(attributes) of the table.	DML adds or updates the row of the table. These rows are called tuples.
DDL statement cannot be rolled back.	DML statement can be rolled back.

1.c)

A database administrator (DBA) is one responsible for directing and performing all activities related to maintaining a successful database environment. A DBA has central access over the DB system and makes sure an organisation's databases and related applications operate functionally and efficiently.

Roles of DBA:

1) Schema definition
2) Storage structure and access method definition
3) Schema and physical-organisation modification
4) Granting of authorisation of data access
5) Routine maintenance

2.a)

1) DDL Interpreter: which interprets DDL statements and records the definitions in the data dictionary.

2) DML compiler: which translates DML statements in a query language into an evaluation plan consisting of low-level instructions that the query evaluation engine understands.

3) Query evaluation engine: which executes low-level instructions generated by the DML compiler.

2.b)

1) Simple attributes: Simple attributes are those that cannot be further divided into sub-attributes. Ex: A student's roll ID.

2) Composite attributes: Composite attributes are made up of two or more simple attributes. Ex: Address

3) Single-valued: Single-valued attributes can only have one value.
Ex: Social security number.

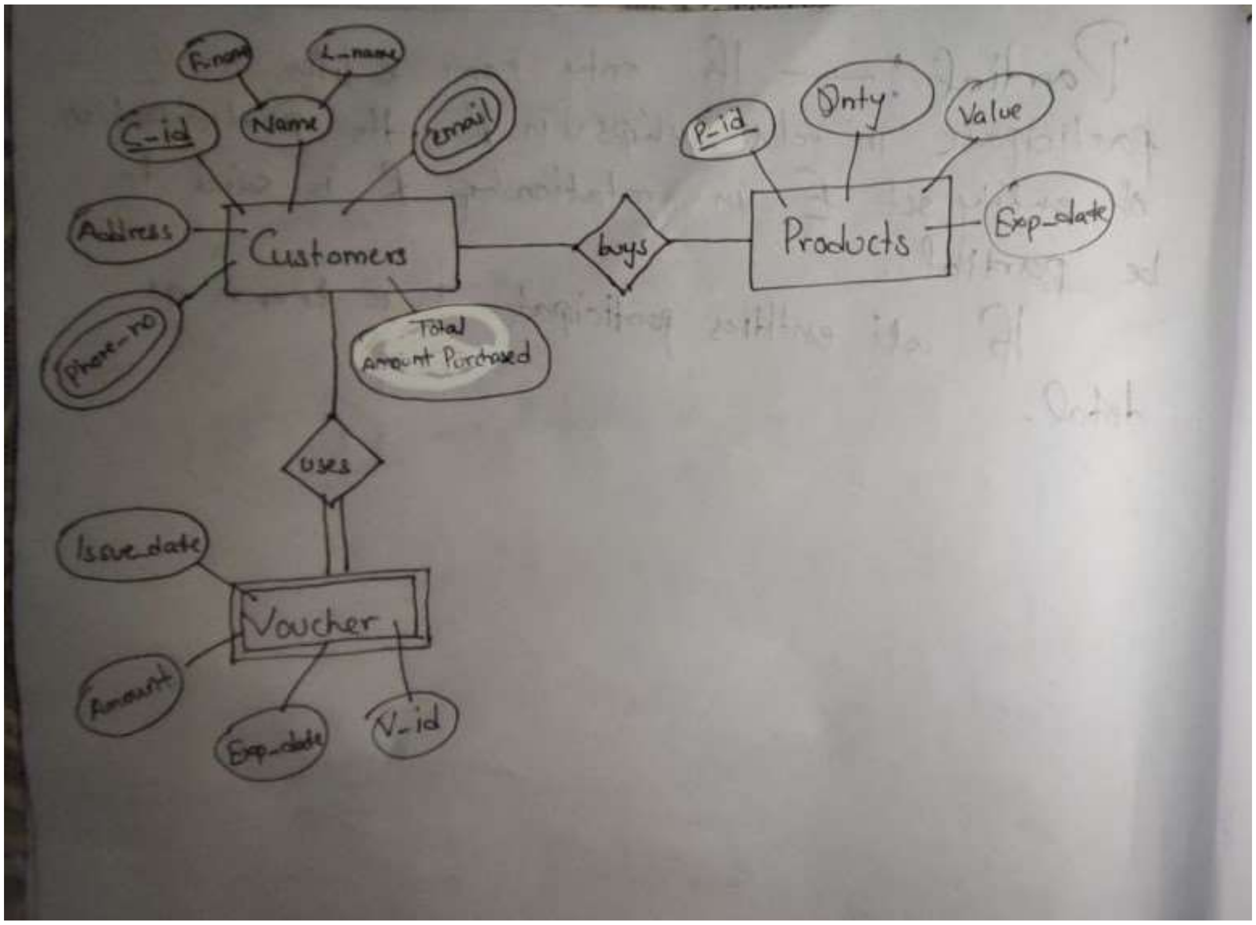
4) Multi-valued: Multivalued attributes can have more than one value.
Ex: Phone number.

5) Derived attributes: Derived attributes are based on other attributes and are not stored directly in the database. Ex: Age

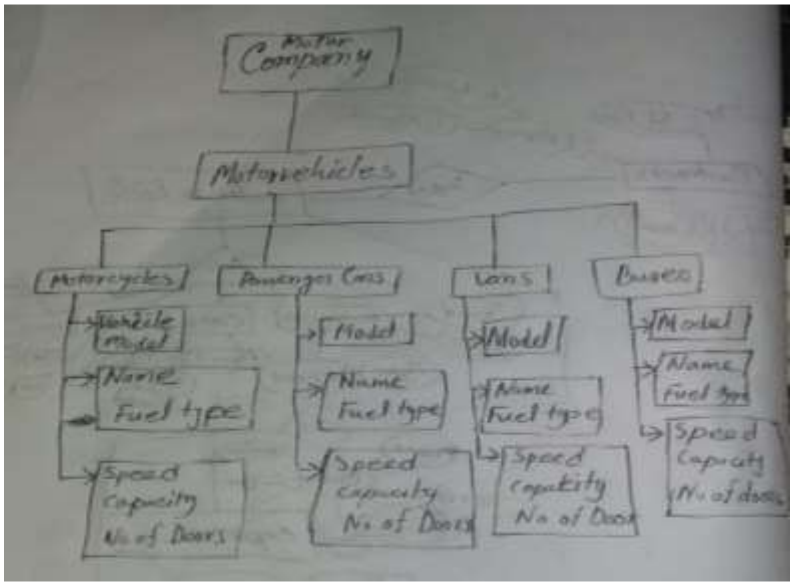
6) Stored attributes: stored attributes are the data that remain constant and fixed for an entity instance. Ex: consider a customer entity in a bank. The customer's name, age, and address would be stored attributes.

7) Null attribute: The value can be absent because it is unknown,
not yet supplied, or nonexistent. Ex: Number of available books.

2. c)



3. a)



Prepared by – Al Fahad & Abdur Rahman

Justification:
• Vehicle model is the most general attribute and should be placed at the top of the hierarchy. It is the broadest level and contains the most general information.
•Name, Horsepower, and Fuel Type should be placed at the second level of the hierarchy. These attributes are common to all vehicle types, but do not provide specific details about any one type.
• Speed, Capacity, and Number of Doors should be placed at the third level of the hierarchy. These attributes are more specific and provide more details about each type of vehicle.
• The attributes should not be placed at a higher or lower level, as this would not accurately represent the hierarchy of the company.

3(b)

An ER diagram can be mapped to a database by creating tables that correspond to the entities or objects in the diagram. Each table should have columns that correspond to the attributes of the entity. For example, if the ER diagram contains an entity called "customer" with attributes "name," "email," and "phone number," then the database should contain a table called "customer" with columns "name," "email," and "phone number." The table should also have a unique identifier for each customer, such as a "customer_id" column. The ER diagram should also contain relationships between entities, which can be mapped to foreign keys in the database. For example, if the ER diagram contains a relationship between "customer" and "order" entities, then the "order" table should have a column called "customer_id" which references the "customer_id" column in the "customer" table.

3(c)

Primary key: The **primary key** of a relational table uniquely identifies each record in the table.

Example:

The ideal primary key for a table of students would be their **ID number**, as this would uniquely identify each student in the table.

Super key: A super key is a set of one or more keys that are used to identify data or records uniquely in a database table.

Example:
Email: If each customer's email address is unique, the Email attribute can be a superkey.
First_Name and Last_Name: If no two customers have the same first and last name, the combination of First_Name and Last_Name attributes can be a superkey.

Candidate key: A super key such that no proper subset is a super key within relation.

Example:
Emp_Id: An attribute that stores the value of the employee identification number.

Prepared by – Al Fahad & Abdur Rahman

Emp_name: An attribute that stores the name of the employee holding the specified employee id.
Emp_email: An attribute that stores the email id of the specified employees.

4(a)

The fundamental operations of relational algebra are used to query relational databases and manipulate data. These operations include:

• Selection (σ): This operation selects a subset of tuples from a relation based on a given condition. For example, the condition “salary > 50,000” can be used to select all employees with a salary greater than 50,000.

• Projection (π): This operation selects a subset of attributes from a relation. For example, you can use the projection “name, salary” to select only the name and salary of employees from a relation.

• Join (⋈): This operation combines two or more relations by matching attributes in each relation. For example, you could join a “customers” relation with an “orders” relation on the “customer_id” attribute to get a list of all orders made by each customer.

• Union (∪): This operation combines two relations and returns the combination of all tuples in both relations. For example, you could use the union operation to combine two relations containing customer information to get a single relation containing all customers.

• Intersection (∩): This operation returns the intersection of two relations, i.e. the set of tuples that are common to both relations. For example, you could use the intersection operation to find customers who appear in both the “customers” and “orders” relations.

• Difference (Δ): This operation returns the difference of two relations, i.e. the set of tuples that are present in one relation but not the other. For example, you could use the difference operation to find customers who appear in the “customers” relation but not the “orders” relation.

• Cartesian Product:
The cartesian product operation is used to combine all rows from one table with all rows from another table. The resulting table will contain all possible combinations of rows.

• Rename
The rename operation in relational algebra is used to change the name of a relation or to rename the attributes of a relation. It is denoted by the Greek letter ρ (rho).
The rename operation can be used to rename a relation. In this form, the operation changes the name of the relation to a new name.
Syntax: ρ(new_relation_name)(old_relation_name)
Example: Suppose we have a relation named "Students" that contains attributes "Name", "Age", and "Gender". We can use the rename operation to change the name of the relation to "Class" using the following syntax:
ρ(Class)(Students)

Prepared by –
Al Fahad
Abdur Rahman