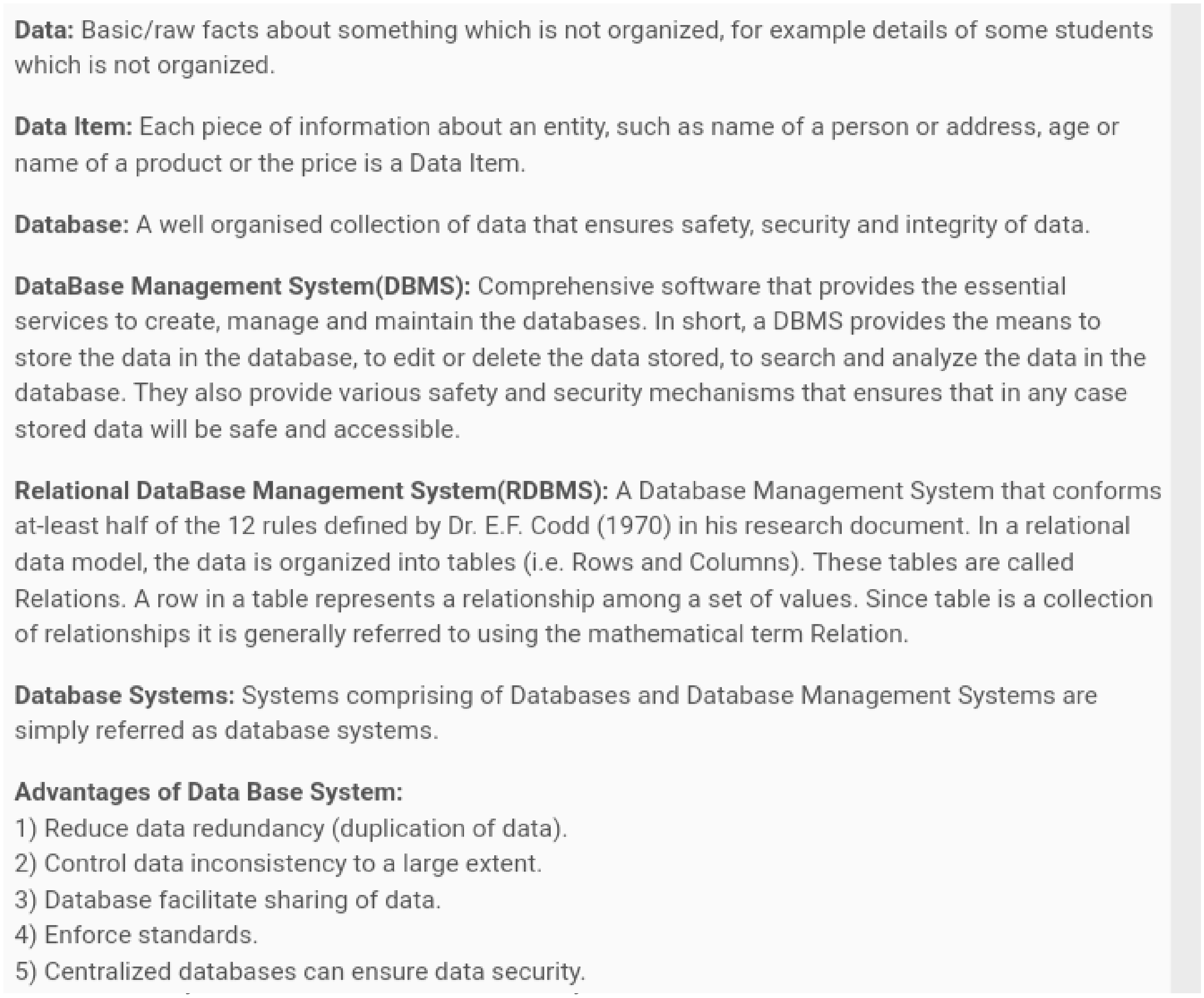
Relational DataBase Management System(RDBMS): A Database



Data: Basic/raw facts about something which is not organized, for example details of some students which is not organized.

Data Item: Each piece of information about an entity, such as name of a person or address, age or name of a product or the price is a Data Item.

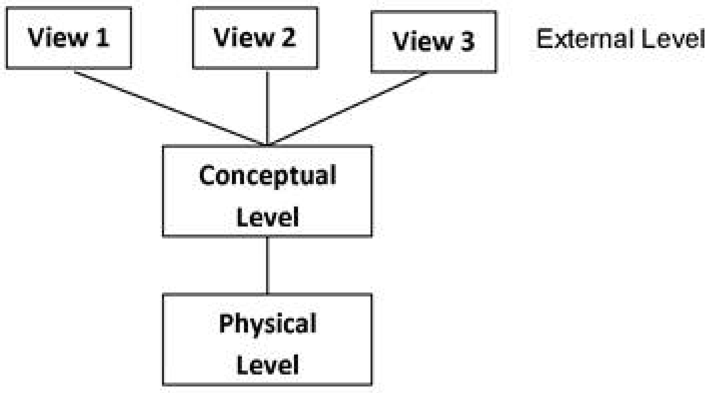
Database: A well organised collection of data that ensures safety, security and integrity of data.

DataBase Management System(DBMS): Comprehensive software that provides the essential services to create, manage and maintain the databases. In short, a DBMS provides the means to store the data in the database, to edit or delete the data stored, to search and analyze the data in the database. They also provide various safety and security mechanisms that ensures that in any case stored data will be safe and accessible.

Relational DataBase Management System(RDBMS): A Database Management System that conforms at-least half of the 12 rules defined by Dr. E.F. Codd (1970) in his research document. In a relational data model, the data is organized into tables (i.e. Rows and Columns). These tables are called Relations. A row in a table represents a relationship among a set of values. Since table is a collection of relationships it is generally referred to using the mathematical term Relation.

Database Systems: Systems comprising of Databases and Database Management Systems are simply referred as database systems.

Advantages of Data Base System:
1) Reduce data redundancy (duplication of data).
2) Control data inconsistency to a large extent.
3) Database facilitate sharing of data.
4) Enforce standards.
5) Centralized databases can ensure data security.



Data Independence: The ability to modify a scheme definition in one level without affecting a scheme definition in the next higher level.

Two Level of Data Independence:
1. Physical Data Independence: It refers to the ability to modify the scheme followed at the physical level without affecting the scheme followed at the

conceptual level.
2. Logical data Independence: It refers to the ability to modify the scheme followed at the conceptual level without affecting the scheme followed at the external level.

Data Model: A way by which data structures and their relationships are analyzed.

Different Data Models:
1. Relational data model
2. Network data model
3. Hierarchical data model

Relational data model: In this model data is organized into tabular structures called relations. A database may contain many relations providing a better classification of data based on its nature and use. Multiple relations are then linked/ associated together on some common key data values (foreign key).

Network Data Model: In this model data is represented by collections of records and relationships among data are represented by links. A record is collection of fields i.e. attributes, each of which contents only one data value.

Hierarchical data model: In this model records are organized as trees, data is represented by collection of records connected to one another through links.

Basics of Relational Model Relation: A tabular structure containing data. To be a relation it must satisfy following four conditions:
■ Atomicity: At every row-column intersection (Cell) there must be an atomic value i.e. a value that can not be further subdivided.
■ No duplicity: No two rows of relation will be identical i.e. in any two rows value in at least one column must be different.
■ Ordering of rows is immaterial.
■ Ordering of columns is immaterial.

Tuple: A row in a relation is called a tuple.

Attribute: A column in a relation is called an attribute.

Domain: Domain of an attribute refers to the set of all the possible values for that attribute.

Degree: Number of attributes in a relation is the degree of that relation.

Cardinality: Number of tuples in a relation is the cardinality of that relation.

Candidate Key: A set of one or more minimal attributes used to uniquely identify a tuple in the relation and which can act as Primary Key. A relation can have multiple candidate keys.

Primary Key: A candidate key that is primarily chosen for unique identification of tuples in a Relation.
Any subset of Primary key should not be Primary key.

Alternate Key: Candidate keys that not chosen as primary key are the alternate keys.

Example: In a LIBRARY Table

Candidate keys can be Accession no, Book no.

Primary key: If we select Book no as primary key for our purpose then

Alternate Key will be Accession No.

Views: A view is a virtual table whose contents are taking from other tables depending upon a condition.

Table: Student

Roll. No.	Name	Marks
101	Anu	85
102	Riya	70
103	Ankit	78

Definition of the VIEW:
CREATE VIEW toppers AS
SELECT * FROM Student
WHERE Marks > 75;
Here name of the view is toppers
Base table is students
toppers(A virtual table based on Student table)

Roll. No.	Name	Marks
101	Anu	85

102	Ankit	78

INTRODUCTION TO MYSQL

MySQL: It is an Open Source RDBMS Software that uses Structured Query Language. It is available free of cost. Key Features of MySQL:

1. High Speed.
2. Ease of Use.
3. Available Free of Cost.
4. Supports standards based SQL.
5. Provides portability.
6. High Security.
7. Provides many data types.
8. Handles large database.

MySQL Data Types: Every column (or data item) should belong to a unique domain (known as data type). These data types help to describe the kind of information a particular column holds. MySQL supports the ANSI SQL data types. Some of the commonly used data types along with their characteristics are as follows:

Class	Data Type	Description	Example
Text	CHAR(size)	A fixed-length string between 1 and 255 characters in length right-padded with spaces to the specified length when stored. Values must be enclosed in single quotes or double quotes.	‘Maths’ ‘TexT’
	VARCHAR(size)	A variable-length string between 1 and 255 characters in length; for example VARCHAR(25). Values must be enclosed in single quotes or double quotes	‘Computer’ ‘Me and u’
NUMERIC	DECIMAL(p,s)	It can represent number with or 17.3 without the fractional part. The size argument has two parts:	20.1 50000.00

		precision and scale. Precision (p) indicates the number of significant digits and scale (s) maximum number of digits to the right of the decimal point	
	INT	It is used for storing integer values	345
Date	DATE	It represents the date including day, month and year between 1000-01-01 and 9999-12-31	2009-07-02

The Structured Query Language(SQL)
SQL (pronounced SEQUEL for Simple English Query Language) is Non-procedural universal data access language used to access and manipulate data stored in nearly all the data bases available currently. SQL standards are defined by ANSI (American National Standards Institute). SQL statements are used to retrieve and update data in a database. SQL works with database programs like MySQL, MS Access, DB2, Informix, MS SQL Server, Oracle, Sybase, etc. Most of the SQL database programs also have their own proprietary extensions in addition to the SQL standard.

SQL Commands
SQL commands can be classified into the following:

Data Definition Language (DDL): A database scheme is defined by set of definitions, which are expressed, by a special set of commands called Data Definition Language (DDL). They are used to create tables, databases, identify data items, provide unique names to the data items and to define the length and provide the range of values that each data item can assume. They are CREATE TABLE, ALTER TABLE and DROP TABLE commands.

Data Manipulation Language (DML): The data manipulation language (DML) handles operations such as entering rows into a table, changing data, deleting rows, and extracting data from rows and tables. With DML, one does not change the table’s structure, but rather its contents. It contains commands like INSERT, UPDATE and DELETE.

Transaction Control Language (TCL): A transaction is a one complete unit of work. A transaction is successfully completed if and only if all its constituent steps are successfully completed. To manage and control the transactions, the transaction control commands are used. e.g. COMMIT, ROLLBACK, SAVEPOINT.

WORKING WITH SQL
To work on MySQL, you need to open or create the database first:
To Create/Open Database:
mysql> CREATE DATABASE <name of database>;
Now the database with the given name will be created. One must be connected to the database before using it, as below:
mysql> use <name of database>;
Creating Tables
Tables are defined with the CREATE TABLE command. When tables are created its columns are named, data types and sizes supplied for each column. At least one column must be specified.
Syntax:
CREATE TABLE <TableName>(<ColumnName1> <Data Type1>, <ColumnName2> <Data Type2>,…….,<ColumnNameN> <Data Type N>);
Example:

```
mysql> CREATE TABLE Students
(
RollNo DECIMAL(3),
Name VARCHAR(25)
);
```

Once the table is created we can insert the record in it, edit or delete existing records, and also we can search for desired record in a very comprehensive way using the SQL Select statement.

Creating tables with SQL Constraints:
^ A Constraint is a condition or check applicable on a field or set of fields.
^ Data constraints are the rules that are defined when a table is created.
^ They can also be defined or modified after creating the tables.
^ When constraints are defined any data entering in the table is first checked to satisfy the condition specified in particular constraint if it is, only then table data can be updated. If data updation/ insertion is violating the defined constraints, database rejects the data (entire record is rejected).
^ When a constraint is applied to a single column, it is called a column level constraint but if a constraint is applied on a combination of columns it is called a table constraint.

Following Constraints can be defined on a table in SQL:

Constraints name	Description
PRIMARY KEY	Used to create a primary key.

UNIQUE	to create a unique key.
NOT NULL	to define that column will not accept null values.
FOREIGN KEY/ REFERENCES	to define referential integrity with another table.
DEFAULT	to define the columns default value.
CHECK	to define the custom rule.

Not Null and Default constraints can be applied only at column level rest all constraints can be applied on both column level and table levels.
Use of constraints:
CREATE TABLE student (Srollno integer NOT NULL, …);
CREATE TABLE student (Srollno integer UNIQUE, …);
CREATE TABLE student (Srollno integer NOT NULL, Sclass integer DEFAULT 12, Sname varchar(30));
CREATE TABLE student (Srollno integer CHECK (Srollno>0), Sclass integer, Sname varchar(30));
CREATE TABLE student (Srollno integer NOT NULL PRIMARY KEY, Sclass integer, Sname varchar(30));
CREATE TABLE teacher (Tid integer NOT NULL, FOREIGN KEY (Studentid) REFERENCES student (Sid));
Inserting the record in existing table:
The INSERT INTO command append a new record to an existing table and initializes it to desired values.
Syntax:
INSERT INTO table_name (column_name [,column_name])
VALUES (value [,value]);
INSERT INTO Student (RollNo, Name)
VALUES (12333,'Anu');
Inserting NULL Values:
INSERT INTO Student (RollNo, Name, Class, Grade)
VALUES (12333,'Anu',11, NULL);
Inserting Dates:
INSERT INTO Student (RollNo, Name, Class, DOB)
VALUES (12333,'Anu',11, '1998-02-24')
Inserting Data from another Table:
INSERT INTO
Marks SELECT * FROM Student
WHERE Class>10;

Note: Column names can be omitted if the values are entered in the same order in which they appear in the table. Insert into will give you an error if you omit to enter a mandatory value (non-null).

Deleting Existing records from the table:
The DELETE command deletes one, many, or even all records in a table, depending on the conditions that you specify.
Syntax:
DELETE FROM tablename
WHERE search_conditions;
for example
DELETE FROM Students
WHERE RollNo>11255;
Note: The delete command is VERY dangerous. If run without conditions, it will delete ALL records in a table. In addition, SQL has no undo function. For instance, DELETE FROM Students;
Will delete all records from Students table. This is not likely to be what you want.

Modifying the contents of records: The UPDATE command changes one, many, or even all records in a table, depending on the conditions that you specify
Syntax:
UPDATE tablename
SET column_name = expression [,column_name = expression..] [WHERE search_conditions];
for example(assuming a customer table)
UPDATE customer SET f_name = 'Thomas'
WHERE l_name = 'Smith' and
date_of_birth = '3/2/1985';
An expression can be either a constant value (e.g., 'Thomas') or an operation done on another column or columns (see the example below, assuming a loan table with column rate.).
UPDATE TABLE loan SET rate = rate + 1.5;
Because there is no condition (i.e., no WHERE) all records will be updated. All rates will be increased by 1.5.
Selecting data from existing table:
SQL SELECT statement is a comprehensive statement used to search/select records from one or more tables. All the analysis done on a database usually involves some form of select statement.
> Choosing all fields (columns): Use an asterisk (*) to indicate all fields with the select statement:
SELECT *
FROM table_name;

SELECT *
FROM customer;

> Choosing a selected list of fields (columns)
SELECT column_name [,column_name] FROM table_name;
SELECT f_name, l_name, date_of_birth FROM customer;
NOTE: The order in which you list the columns affects their order in the resulting output. Items within [ ] are optional.

> Temporarily renaming columns in query results
SELECT column_heading AS column_name [,column_heading AS column_name] FROM table_name;
Example:
SELECT f_name as “Name”
FROM customer;

> Including calculated columns in the results
SELECT date_due, rate, principal, rate * principal FROM loan;
NOTE: If necessary, use parentheses to clarify order of precedence.

> Eliminating duplicate query results with distinct
If you use the keyword distinct after the keyword SELECT, you will only get unique rows. Example:
SELECT rate,
FROM loan;
(above will display all rate values might be repeated)
SELECT distinct rate FROM loan;
(above will display only unique rate values, no repetition)

> Selecting from all the rows:
SELECT ALL rate,
FROM loan;
(above query will display all rate values)

> Selecting rows: WHERE clause is used to specify the condition for searching. Only those records will be retrieved that satisfy condition given with where clause.
SELECT SELECT_list
FROM table_list
WHERE search_conditions;
Example:
SELECT * FROM customer
WHERE f_name = ‘Carl’;

> Possible Search Conditions
– Comparison operators (=,<,>,!=.<>,<=,>=)
SELECT * FROM loan
WHERE principal > 100000000;

– Ranges (between and not between; inclusive)
SELECT * FROM loan
WHERE rate BETWEEN 7.5 AND 8.5;

OR

SELECT * FROM loan
WHERE rate NOT
BETWEEN 7.5 AND 8.5;

– Lists (in and not in)
SELECT *
from Customer
where city IN (‘Ahmedabad’, ‘Baroda’, ‘Delhi’, ’Mumbai’, ’Chennai’);

OR

SELECT *
from Customer
where city NOT IN (‘Ahmedabad’, ‘Baroda’, ‘Delhi’,’Mumbai’,’Chennai’);

– Null values
SELECT *
from Customer
where city is Null;

OR

SELECT *
from Customer
where city is Not Null;

– Character matches (like and not like)
SELECT f_name, l_name
FROM customer
WHERE l_name LIKE ‘Fos%’;
SELECT f_name, l_name
FROM customer
WHERE l_name LIKE ‘_oster’;
Note: “%” (matches any string of zero or more characters) and “_” (matches any one character). In addition to those, brackets can be used to include

either ranges or sets of characters.
Combinations of previous options using logical operators and, or, and not
etc.: SELECT f_name, l_name FROM customer
WHERE l_name LIKE ‘San%’ AND City NOT IN (‘Baroda’,‘Delhi’)

> Some more examples:
– ‘Am%’ matches any string starting with Am.
– ‘%Singh%’ matches any string containing ‘Singh’
– ‘%a’ matches any string ending with ‘a’
– ‘ ’ matches any string that is exactly 3 characters long.
– ‘ %’ matches any string that has at least 2 characters long.
– ‘ g’ matches any string that is 4 characters along with 3 characters in the beginning but ‘g’ as the 4th character.

> Viewing a tables structures
Describe/ Desc statement is used to see the structure of a table:
Desc <tablename> ;
Describe <tablename>;

> Sorting records
The output of a SELECT query can be sorted in ascending or descending order on one or more columns, the default is ascending. This is important to note that the data in table is not sorted, only the results that appear on the screen are sorted.
Syntax:
SELECT <column name> [,<column name>, ….] FROM <table name>
[WHERE <condition>] [ORDER BY <column name> [, <column name>.]];
Example: (Sorting on single column)
SELECT * FROM EMPL ORDER BY ENAME;
Example: (Sorting on Multiple columns)
SELECT * FROM EMPL ORDER BY ENAME, JOB;

> Adding a column:
The ALTER TABLE command is used to change definitions of existing tables. It can add columns, delete columns or change their size,rename the name of an existing table.
Syntax:
ALTER TABLE <table name>
ADD (<column name> <data type with size> <constraints>);
Example:
ALTER TABLE Students
ADD (age NUMBER (2) CHECK (age > 5));

> Modify a column:
Syntax:

ALTER TABLE <table name>
MODIFY (column name newdatatype (newsize));
Example:
ALTER TABLE Students MODIFY (age NUMBER (1));

> Changing a column name:
ALTER TABLE <table name>
CHANGE <old_column_name> <new_column_name> <column definition> ;
Example:
ALTER TABLE Students CHANGE age s_age NUMBER (2)

> Removing table components
– To remove primary key constraints ALTER TABLE Students DROP primary key;
– To remove column from the table ALTER TABLE Students DROP COLUMN age;
> Drop a table from database:
DROP TABLE <table name>;
Example:
DROP TABLE Students;

> Renaming a table:

ALTER TABLE <old table name>

RENAME TO <new table name>;

Example:

ALTER TABLE Students RENAME TO Students_Details;

Operator Precedence:

All the operators have precedence. Precedence is the order in which different operators are evaluated. Various operators in descending order of precedence (top to bottom) are listed below:

1	!
2	(Unary minus)
3	^
4	*,/,DIV,%, MOD
5	-,+