Unit – 1
Introduction to Database System & SQL Commands

Y.A.HATHALIYA

CO and Topics Covered

CO1: Perform queries on datasets using SQL*Plus

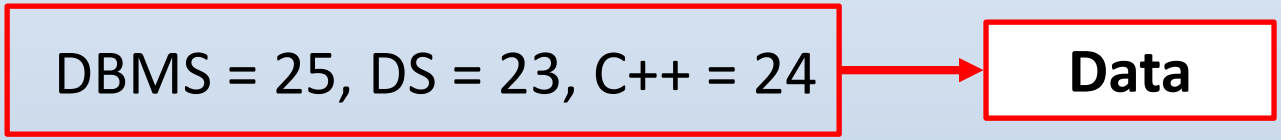
Topics:

- Basic Definition of : Data, Information, Data Item, Records, Files, Metadata, Data dictionary and it's Components, Database and Database Systems and Database Environment.
- Schemas, Sub-Schemas, and Instances.
- SQL Data Types.
- SQL DDL Commands.
- SQL DML Commands.
- SQL TCL Commands.
- SQL DCL Commands.

Data and Information

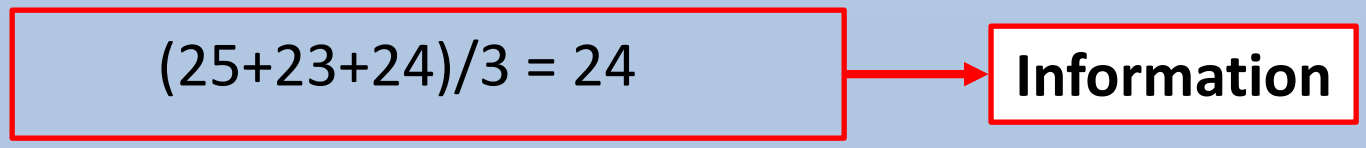
Data

- Data Means known facts, that can be stored.
- For Example, marks of 3 subject are given below, that is known as Data



Information

- Information means processed or organized data
- For Example, if we count the average of 3 subjects then it known as Information



Basic Definitions Related With Database

Database

- It is a collection of inter-related data and It represents some aspect of the real world.
- For example student information database, book bank database, college database etc…

Management

- Manipulation, searching and security of data
- For example searching of product in amazon, viewing result in GTU website etc…

System

- Programs or Tools used to manage database.
- For example SQL Server Studio, Oracle 11g

DBMS (Database Management System)

- DBMS = Database + Set of Programs (that manipulate the data)
- Data manipulation involves various operations like store data, modify data, remove data and retrieve data.
- Some examples of DBMS are:
  - My SQL
  - Oracle
- Operations Performed on DBMS:
  - Creating tables for database.
  - Inserting new data into existing database.
  - Modifying or updating existing database.
  - Removing data from existing database.
  - Deleting or destroying tables or files for database.

Metadata

- It is data about data.
- Data such as table name, column name, data type, authorized user for any table is called metadata for that table.

Faculty

Fac_ID	Name	From	Initial	Mobile
101	Yagnik Hathaliya	Rajkot	NJR	123456
102	Niraj Trivedi	Rajkot	CNK	456789

Table Name – Faculty
Column Name – Fac_ID, Name, From, Initial, Mobile
Data Type – Varchar,Number

Data Dictionary

- It is an information repository which contains metadata.
- This can involves information such as table name, owners, column names, data types, size and constraints.

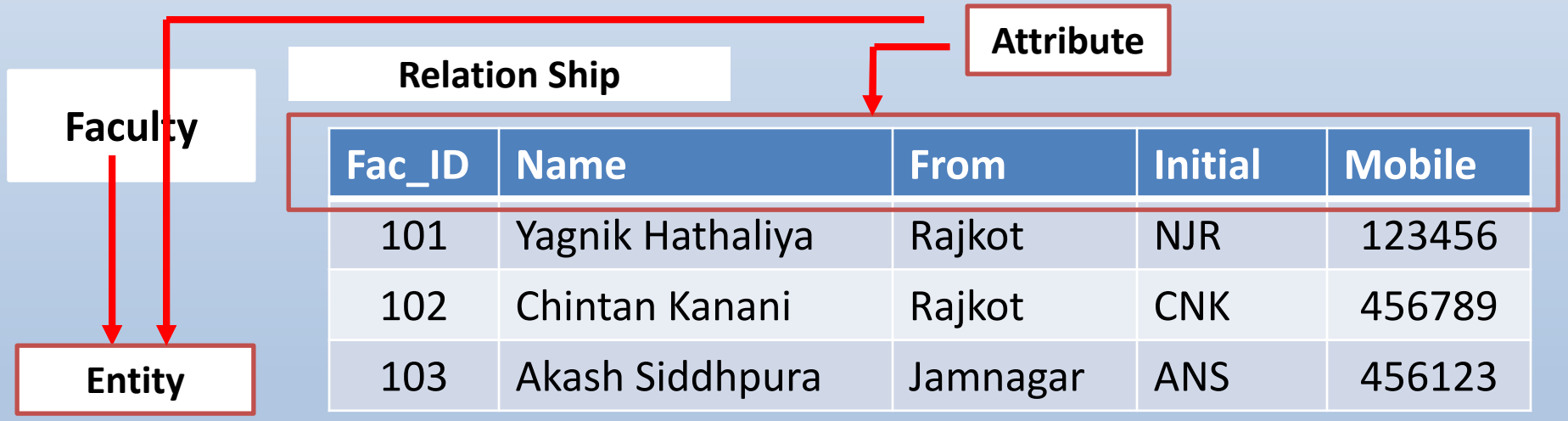
Faculty

Fac_ID	Name	From	Initial	Mobile
101	Yagnik Hathaliya	Rajkot	NJR	123456
102	Chintan Kanani	Rajkot	CNK	456789
103	Akash Siddhpura	Jamnagar	ANS	456123

Table Name – Faculty
Owner – Database User (Admin, Faculty, Student etc…)
Column Name – Fac_ID, Name, From, Initial, Mobile
Data Type – Varchar,Number
Constraints – Not Null, Primary Key, Unique Key etc…

- The Basic Components of Data Dictionaries.

1. Entities – A thing or object or person in the real world, that is different from all other object.
2. Attribute – Property or characteristics of an entity.
3. Relation Ship – It is an association between several entities.
4. Key – A data item or a field which is used to identify a record in a database is referred as key.



Fac_ID	Name	From	Initial	Mobile
101	Yagnik Hathaliya	Rajkot	NJR	123456
102	Chintan Kanani	Rajkot	CNK	456789
103	Akash Siddhpura	Jamnagar	ANS	456123

Active and Passive Data Dictionary

- If a data dictionary is managed automatically by database management software then it is known as active data dictionary, it is also called as an integrated data dictionary.
- If the user manage it then it is known as passive data dictionary, it is also called as an non-integrated data dictionary.

Data Items (Field)

- It is a character or group of characters (alphabetic or numeric) that have specific meaning.
- It is represented in the database by a value

Record

- It is a collection of logically related fields.
- Here each field in a record contains fix size and fixed data type.

Field

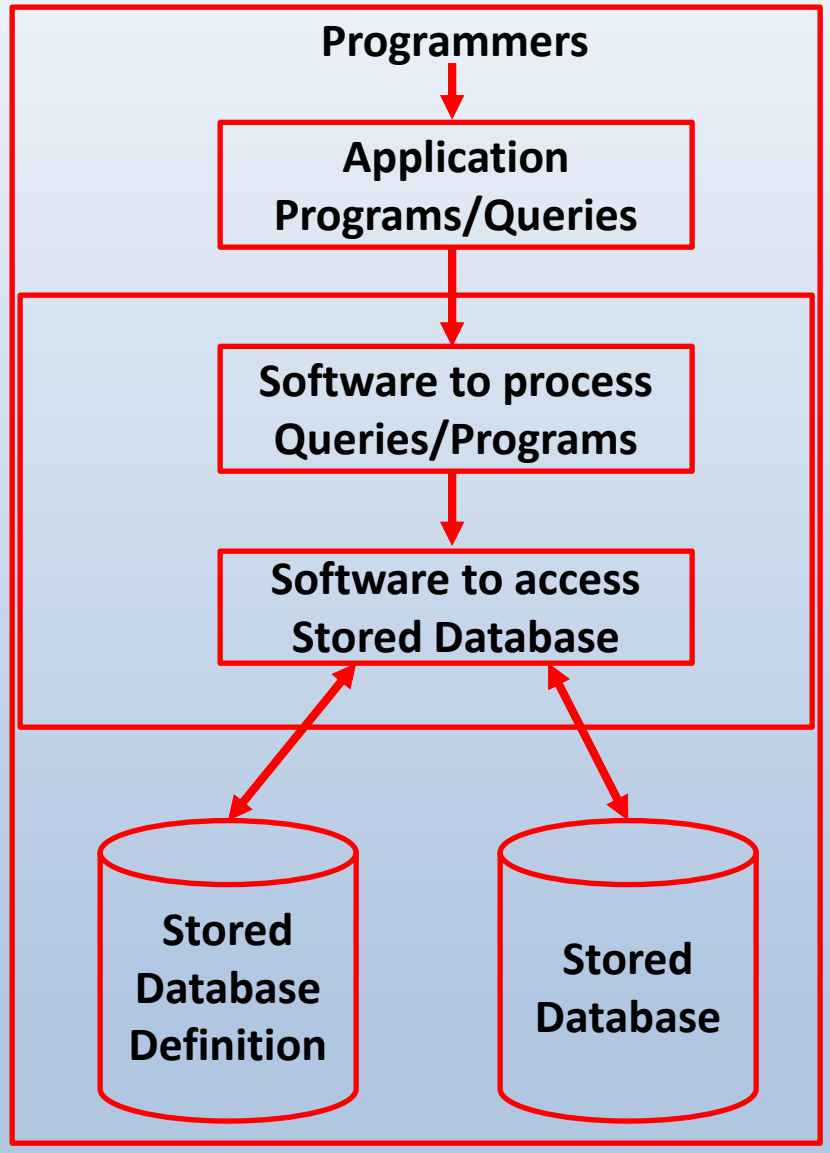
Record

Fac_ID	Name	From	Initial	Mobile
101	Yagnik Hathaliya	Rajkot	NJR	123456
102	Chintan Kanani	Rajkot	CNK	456789
103	Akash Siddhpura	Jamnagar	ANS	456123

Files

- It is collection of related records.
- These records are generally arranged in a specific sequence.

Database System Environment



- A database system is a collection of database and a set of programs to manipulate data stored in a database.
- Database System Environment can be divide in to 3 parts,

1. User
   I. Database Administrator
   II. Database Designer
   III. Application Programmers
   IV. End Users
2. Software
3. Hardware

1. Hardware

- All the physical devices of a computer system are referred as hardware.
- A computer system can have number of different hardware such as processor, memory, hard disk, monitor, keyboard, mouse etc.
- From the database point of view hardware can be divided into 2 categories;

I. <u>The processor and main memory</u>
   - Supports the execution of the database software

II. <u>The secondary storage devices</u>
   - Used to store data of system permanently
   - These includes hard disks, magnetic tapes, compact disks, pen drive etc…

2. Software

- Software provides the interface between users and database stored in physical devices.
- Application program, DBMS and OS together generate the software component here.
- Application program are developed using programming languages like C, C++, Java and etc…
- These application programs use the functionalities of the DBMS software to perform various operations on the database.
- The OS (Operating System) manages all hardware of the computer.

3. Users

- Any person who communicate with the database is known as Database User.

I. Database Administrator

- Has central control over the database system including data & programs.
- DBA is responsible for proper functioning of the database system.

II. Database Designer

- Identify the data to be stored in database and designs structure of the database for an organization.

III. Application programmers

- These users write application programs to interact with the database.
- Application programs can be written in some programming languages such as C++, Java, .Net etc…
- Such programs access the database by issuing the request, typically a SQL statement to DBMS.

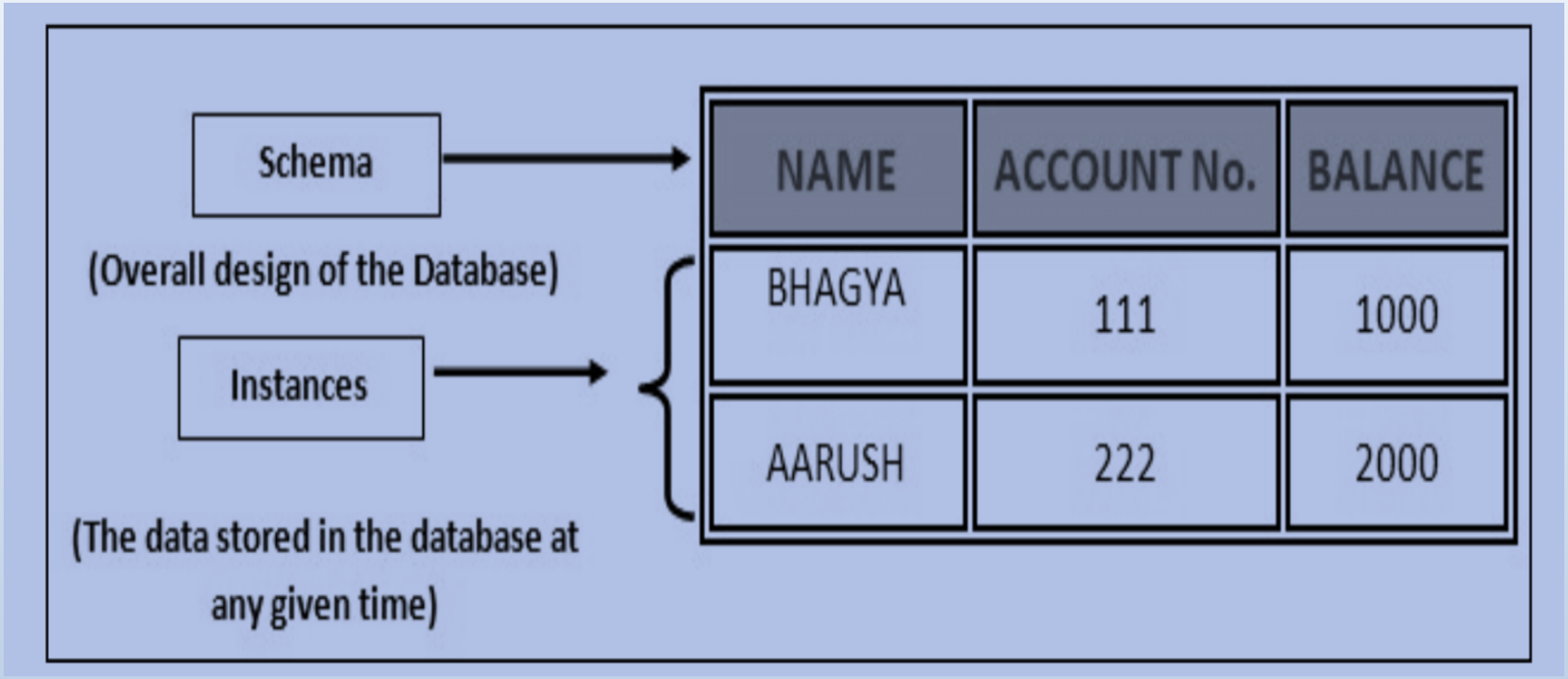
IV. End Users

- Interact with the system by using pre-developed application programs.
- End users not know about the working, database design, the access mechanism etc.
- They just use the system to get their task done.

Applications of DBMS

- DBMS is computerized record-keeping system, so where it is required to store some data at there DBMS can be used.
  - E-Commerce (Flikart, Amazon, Shopclues etc…)
  - Online Television Streaming (Hotstar, Amazone Prime, etc…**)**
  - Social Media (WhatsApp, Facebook, Twitter etc.)
  - Banking & Insurance
  - Airline & Railway
  - Universities and Colleges
  - Human Resource
  - Hospitals
  - Government Organizations

Schema and Instances



Schema	Instance
Overall logical design of the database	Collection of information stored in the database
It includes table name, column name, data types and size of the columns, various constraints	Actual data or information stored in table in form of different records or row
It is not changed frequently	Changes frequently
For example, Create/Drop of table or columns, changes in data types, size or constraints on any column	For example, Insert, delete or update operation on database
Sub Schema: it is just a plan or schema for the view purpose only which refers to users' view.	

File System	Database System
- It coordinates only physical access of data.	- It coordinates both physical and logical access to data.
- It has predetermined access of data (i.e. compiled programs)	- It has flexible access of data (i.e. queries)
- It stores unstructured data.	- It stores structured data.
- No concurrent access for the same time.	- Multiple users can access the same data at same time.
- Low security in it.	- High security in it.
- There should be integrity problems.	- Not such problems due to integrity constraints.
- Data redundancy is there.	- No data redundancy.
- Data is accessed through single or various files.	- Tables (schema) are used to access data.
- Data Inconsistency is more in file system	- Data Inconsistency is less in database system
- It is less complex	- It is more complex
- It is more flexible	- It is less flexible

Data types

Numerical Data type

- Each column in a database table is required to have a name and a data type.
- It is used to store zero, negative and positive values.

No	Data type	Description
1	number	Floating-point number (size and precision depends on the database we are using)
2	integer / numeric / decimal	Fixed-point number (whole number) with precision of 38 (default size)
2	number (p)	Fixed-point number (whole number) with a scale zero and precision p
3	number (p, s)	Floating-point number P refers to precision S refers to scale
4	float / real	Floating-point numbers with varying precision.

Character/String Data type

- There are four different data types available to store character or string values, as described below:

No	Data type	Description
1	char(size)	▪ Stores character string of fixed length. ▪ Size represents the number of characters (length) to be stored (default size is 1). ▪ Maximum length size is 255 characters (bytes).
2	varchar(size) varchar2(size)	▪ Stores characters of variable length. ▪ It is more flexible than 'char' type. ▪ No default size is available, so you must specify size. ▪ Maximum length is 2000 characters.
3	Long	▪ Stores large amount of character strings of variable length. ▪ Maximum length is up to 2 GB. ▪ Only one column per table can be defined as a 'long'. ▪ A 'long' column cannot be used with WHERE, ORDER BY or GROUP BY clauses.

Date Data type

- It is used to store date and time.
- The standard format is DD-MON-YY to store date.
- The current date and time can be retrieved using function SYSDATE

Binary Data type

- Generally images, audio and video files are comes under the type of binary data type.

No	Data type	Description
1	RAW	▪ Stores binary type data. ▪ Maximum length is up to 32767 bytes.
2	LONG RAW	▪ Stores large amount of binary type data. ▪ It is often referred as Binary Large Object (BLOB). ▪ Maximum length is up to 2 GB.

Components of SQL

- SQL = **S**tructured **Q**uery **L**anguage
- It is a standard language for access and manipulate databases.
- Features of SQL:
  - Non procedural Language
  - English Like Language
  - Open Source
  - Non Case Sensitive Language
- What SQL can do:
  - create new databases
  - create new tables in a database
  - execute queries against a database
  - retrieve data from a database
  - insert records in a database
  - update records in a database
  - delete records from a database

- Rules for SQL:
  - Statements start with verb. For example – SELECT, CREATE, DESCRIBE etc.
  - A semicolon (;) is used to end SQL statements.
  - A comma (,) is used to separate parameters.
  - A space separates keywords, like DROP TABLE PERSON;
  - Characters and date constants must be enclosed within single quote (‘ ’). Like ‘abc’ or ‘10-JAN-15’ etc.
  - No need of single quote in number values.

- Components of SQL
  1. **DDL (**Data Definition Language)
  2. **DML (**Data Manipulation Language)
  3. **DQL (**Data Query Language)
  4. **DCL (**Data Control Language)
  5. **TCC (**Transactional Control Commands)

DDL (Data Definition Language)

- A set of SQL commands to change the structure of the table.
- It simply deals with descriptions of the database schema.
- Used by DBA or Database designer.

CREATE	**It is used to create table or database and its objects (index, function, views, store procedure and triggers)**
ALTER	**To alter the schema or logical structure of the database or table.**
TRUNCATE	**To remove all the records from the table**
DROP	**To delete the tables from the database**
RENAME	**To rename the existing objects of database.**

DML (Data Manipulation Language)

- A set of SQL commands to change the data of the table.
- It allows users to insert, update, delete data from the database and its objects.

INSERT	**To insert data into a table**
UPDATE	**To modify existing data in a table**
DELETE	**To delete records from the table**

DQL (Data Query Language)

- Allows data retrieval from the database.

SELECT	Allows data retrieval in different ways from a table.

DCL (Data Control Language)

- Set of SQL commands used to control access to data and database
- DCL commands are grouped with DML commands.

GRANT	**To give access privileges to users on the database**
REVOKE	**To withdraw access privileges given to users on the database**

TCC (Transactional Control Commands)

- Managing the changes affecting the data.

COMMIT	**To save work permanently**
ROLLBACK	**It restores the database to original state since the last COMMIT.**
SAVEPOINT	**It identifies a point in a transaction to which you can later roll back.**

Create Database (Create Database Query)

- The Create Database statement is used to create a new SQL Database.
- The basic syntax is,

create database	Database name	;

- The Example is,

```
create database avpti;
```

Use Database (Use Database Query)

- The Use Database statement is used to use a particular SQL Database.
- The basic syntax is,

Use database	Database_name	;

- The Example is,

```
use database avpti;
```

Drop Database (Drop Database Query)

- The Drop Database statement is used to drop or delete an existing SQL Database.
- The basic syntax is,

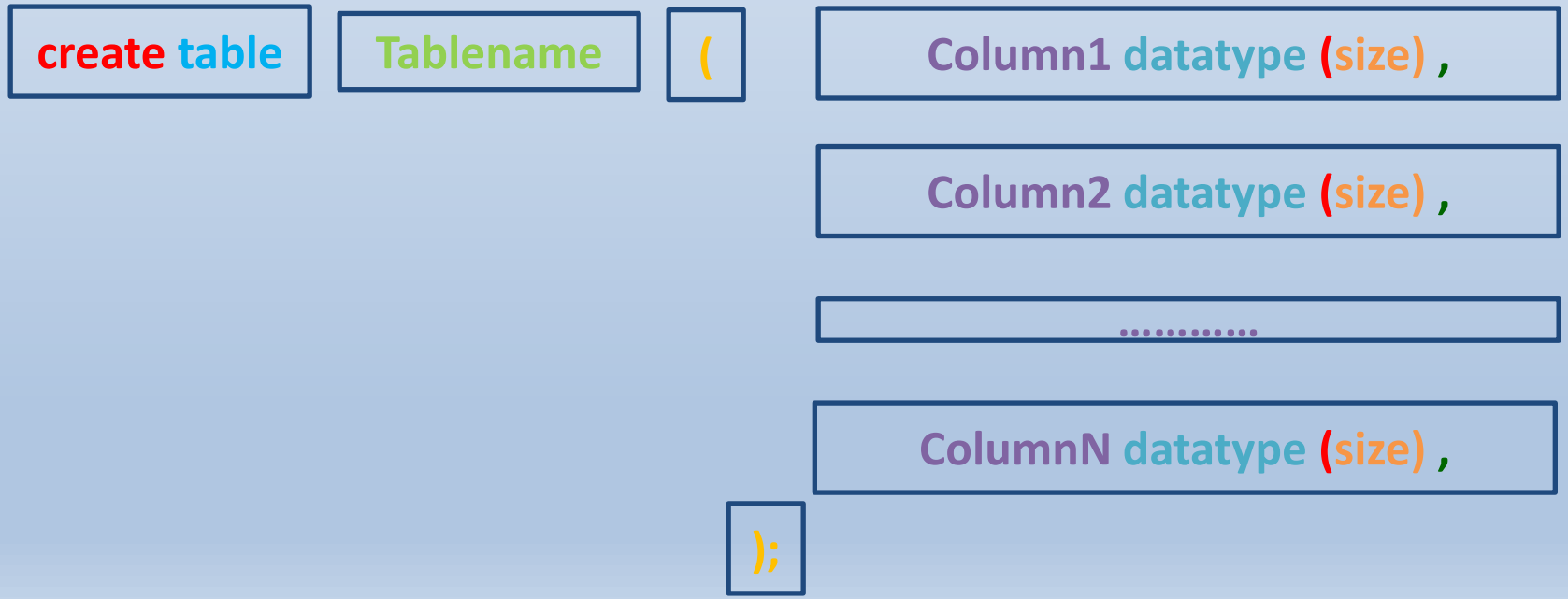
```
drop database   Database_name   ;
```

- The Example is,

```
drop database avpti;
```

Create Table (Create Table Query)

- The Create Table statement is used to create a new table in a Database, remember table name must be unique.
- Each column definition requires name, data type and size for that column, Each column definition is separated from other by a ‘,' (comma), The entire SQL statement is terminated with ' ; ' (semi colon).
- The basic syntax is,



Example:

Create a 'test' table having three columns (id, name and age) described below:

test table →

Column name	Data type	Size
id	Number	5
name	Varchar2	15
age	Number	2

Input command →

Output →

```
SQL> CREATE TABLE test (id number(5), name varchar2(15), age number(2));

Table created.

SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ----------------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(2)
```

Describe Table (Describe Table Query)

- The Describe Table statement is used to verify whether a table has been created according to specification or not? Means it displays a logical structure of your table.
- The basic syntax is,

desc	Tablename	;

Example (*Show table schema of a table 'test'*)

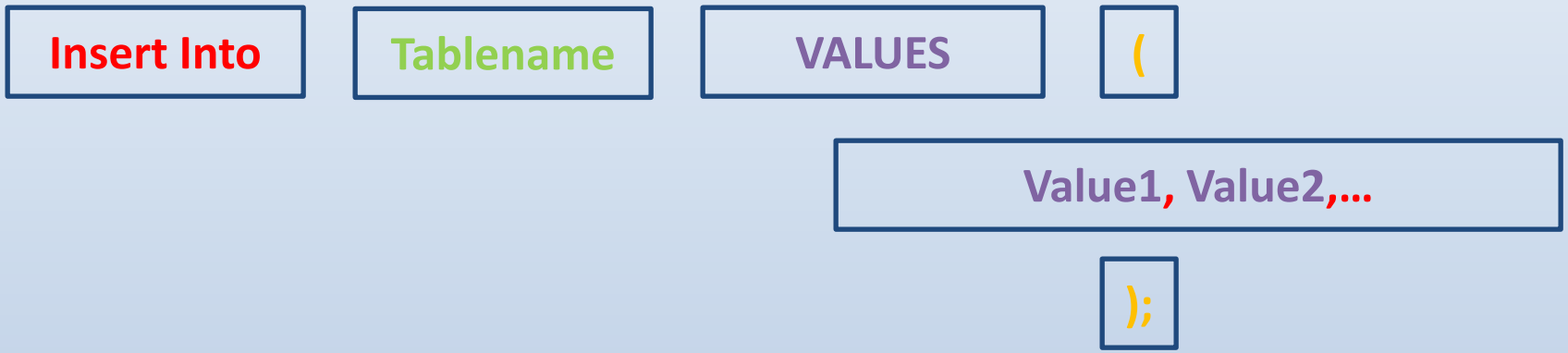
Input command →

Output →

```
SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(2)
```

Insert Data into a Table (Insert Query)

- The Insert into statement is used to insert new records in a table that has been created.
- The basic syntax is,



Example (*Insert data in the table test*)

```
SQL> INSERT INTO test(id, name, age)
  2  VALUES (1, 'BHAGYA', 15);

1 row created.

SQL> INSERT INTO test VALUES(2, 'AARUSH', 10);

1 row created.

SQL> SELECT * FROM test;

        ID NAME                           AGE
---------- ---------------- ----------
         1 BHAGYA                          15
         2 AARUSH                          10
```

Select Data from the Table (Select Query)

- The select statement is used to fetch the data from a database table that has been created earlier.
- The basic syntax is,

→**Display all rows and all columns from table**

```
select * from Tablename ;
```

Example (*Display all the records from the table 'test'*)

```
SQL> SELECT * FROM test;

        ID NAME                          AGE
---------- --------------- ----------
         1 bhagya                        15
         2 aarush                        10
         3 shreeja                        7
         4 kahaan                         3
```

→Selected Row But All Columns

select	*	from	Tablename	where	condition	;

Example (*Display only those records that have age greater than 10*)

```
SQL> SELECT * FROM test WHERE age > 10;

        ID NAME                           AGE
---------- ---------------- ----------
         1 bhagya                        15
```

→Selected column, All Rows

select	column1	column2	.....	column n	from	Tablename;

Example (*Display only name and age of all the records from the table 'test'*)

```
SQL> SELECT name, age FROM test;

NAME                          AGE
--------------- ----------
bhagya                         15
aarush                         10
shreeja                         7
kahaan                          3
```

→Selected Column, Selected Rows

select	column1	column2	.....	column n	from	Tablename

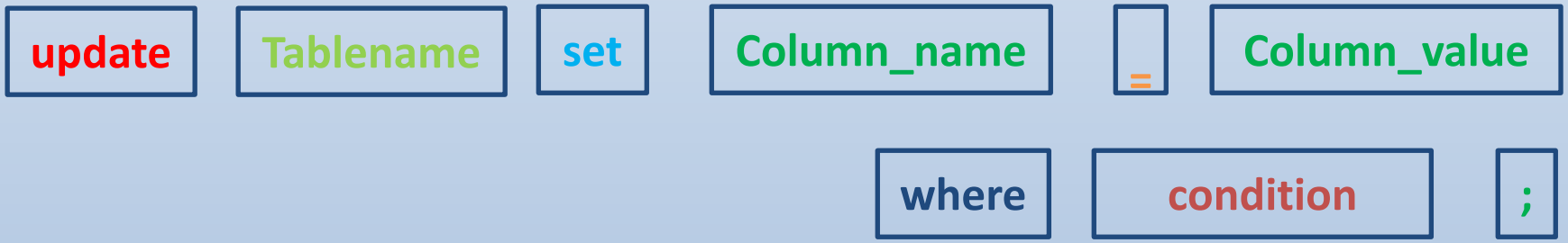
where	condition	;

Example (*Display only name whose age greater than 10*)

```
SQL> SELECT name FROM test WHERE age > 10;

NAME
---------------
bhagya
```

Update Data in a Table (Update Query)

- The update statement is used to update records in a table that has been created.
- It updates either all rows or a set of rows from a table.
- The SET clause specifies which column data to modify. Using WHERE, we can update specific record.
- The basic syntax is,

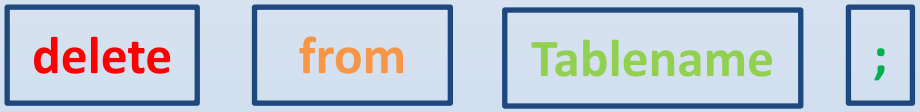


Example (*Change the age of AARUSH in table 'test' from 10 to 15*)

```
SQL> SELECT * FROM test;

        ID NAME                           AGE
---------- --------------- ----------
         1 BHAGYA                          15
         2 AARUSH                          10

SQL> UPDATE test SET age=15 WHERE name='AARUSH';

1 row updated.

SQL> SELECT * FROM test;

        ID NAME                           AGE
---------- --------------- ----------
         1 BHAGYA                          15
         2 AARUSH                          15
```

Delete Data from a Table (Delete Query)

- The delete statement is used to delete records from a table that has been created.
- The basic syntax is,
  - **Delete all records from table**



  - The Example is,

    delete from test;

- **Delete a specific record from table**

delete	from	Tablename	where	condition	;

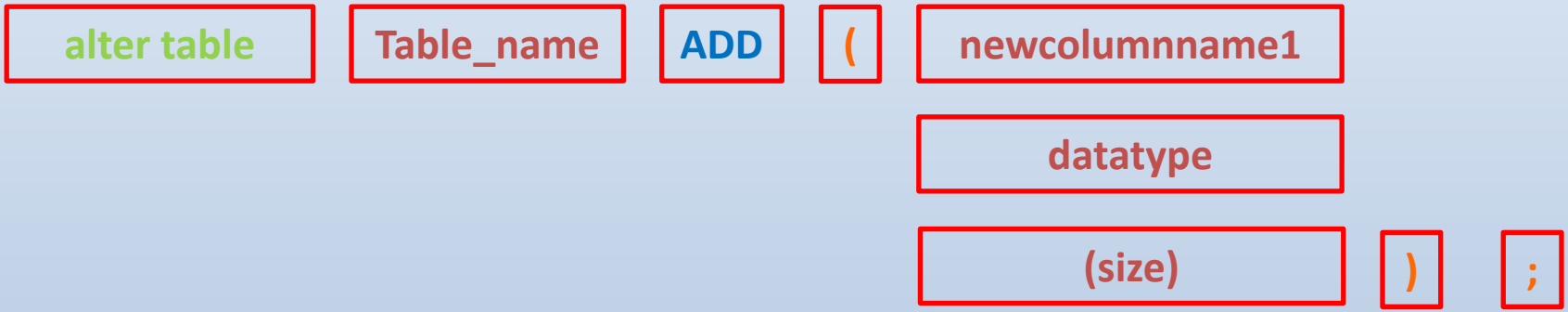
Example (*Delete the record from the table 'test' whose id is 1*)

```
SQL> SELECT * FROM test;

        ID NAME                          AGE
---------- -------------- ----------
         1 BHAGYA                        15
         2 AARUSH                        15
```

```
SQL> DELETE FROM test WHERE id = 1;

1 row deleted.

SQL> SELECT * FROM test;

        ID NAME                          AGE
---------- -------------- ----------
         2 AARUSH                        15
```

Alter Data in a Table (Alter Table Query)

- Alter command used to modify structures of a table.
- Alter command can be used to add ,modify, or drop columns in a table.
- The basic syntax is,
  - **Add new column in to table**



Example (*Add a new column ('gender') to a test table having data type character of size 2)*

Input command →

Output →

```
SQL> ALTER TABLE test ADD gender char(2);

Table altered.

SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(2)
 GENDER                                             CHAR(2)
```

- **Modify existing column in to table**

```
alter table  Table_name  MODIFY  (  New_column_name1
                                    datatype
                                    (size)  )  ;
```

Example (*Change the size of the field 'age' from 2 to 4 in the table 'test'*)

```
SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(2)

SQL> ALTER TABLE test MODIFY (age number(4));

Table altered.

SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(4)
```

- **Drop/Remove column in to table**

alter table	Table_name	DROP COLUMN	Column_name	;

Example (*Remove (or Drop) a column 'gender' from table' test'.*)

```
SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(2)
 GENDER                                             CHAR(2)

SQL> ALTER TABLE test DROP COLUMN gender;

Table altered.

SQL> DESC test;
 Name                                      Null?    Type
 ----------------------------------------- -------- ------------
-
 ID                                                 NUMBER(5)
 NAME                                               VARCHAR2(15)
 AGE                                                NUMBER(2)
```

Input command → `SQL> ALTER TABLE test DROP COLUMN gender;`

Output → `Table altered.`

Truncate Query in a Table

- The truncate statement is used to remove/delete all the records from a table that has been created.
- The basic syntax is,

truncate	Table	Tablename	;

Example:

Delete all the rows from the table 'test'.

Input command →
Output →

```
SQL> TRUNCATE TABLE test;

Table truncated.

SQL> select * from test;

no rows selected
```

Drop Query in a Table

- The drop statement is used to delete selected table (along with their record) from a particular database.
- The basic syntax is,

Remove the table 'test' from the database.

Input command →

Output →

```
SQL> DROP TABLE test;

Table dropped.

SQL> DESC test;
ERROR:
ORA-04043: object test does not exist
```

Rename Query in a Table

- The rename statement is used to rename a table from a particular database.
- The basic syntax is,

```
rename  Old_table_name  to  New_table_name  ;
```

Example:

Rename the table 'test' and set a new name as 'testnew'.

Input command →

```
SQL> RENAME test TO testnew;
```

Output →

```
Table renamed.
```

Commit Command

- The commit command is used to save work permanently in database, that terminates the current transaction and makes all the changes permanent.
- Various data manipulation operations such as insert, update and delete are not effect permanently until they are committed.
- The basic syntax is,

```
COMMIT;
```

Example:

Suppose we have a table 'test' like:

NO	NAME	AGE
1	aarush	5
2	bhagya	10
3	vihaan	4
4	virat	35

Now, if we want to remove the 4th record and save that transaction permanent, we can use commit in following way.

Input command →

Output →

```
SQL> DELETE FROM test WHERE no = 4;

1 row deleted.

SQL> COMMIT;

Commit complete.

SQL> SELECT * FROM test;

        NO NAME                AGE
---------- ---------- ----------
         1 aarush                5
         2 bhagya               10
         3 vihaan                4
```

Rollback Command

- The Rollback command is used to undo work and restore database to previous state.
- It terminates the current transaction and undone any changes made during the transaction.
- Oracle also performs auto rollback. In situation like, Computer failure, Oracle automatically rollbacks any uncommitted work, when the database bought back next time.
- The basic syntax is,

```
ROLLBACK;
```

Saved the transaction using commit. →

Input command →

Output →

```
SQL> SELECT * FROM test;

        NO NAME                 AGE
---------- ---------- ----------
         1 aarush                5
         2 bhagya               10
         3 vihaan                4
         4 virat                18

SQL> COMMIT;

Commit complete.

SQL> DELETE FROM test WHERE no=4;

1 row deleted.

SQL> SELECT * FROM test;

        NO NAME                 AGE
---------- ---------- ----------
         1 aarush                5
         2 bhagya               10
         3 vihaan                4

SQL> ROLLBACK;

Rollback complete.

SQL> SELECT * FROM test;

        NO NAME                 AGE
---------- ---------- ----------
         1 aarush                5
         2 bhagya               10
         3 vihaan                4
         4 virat                18
```

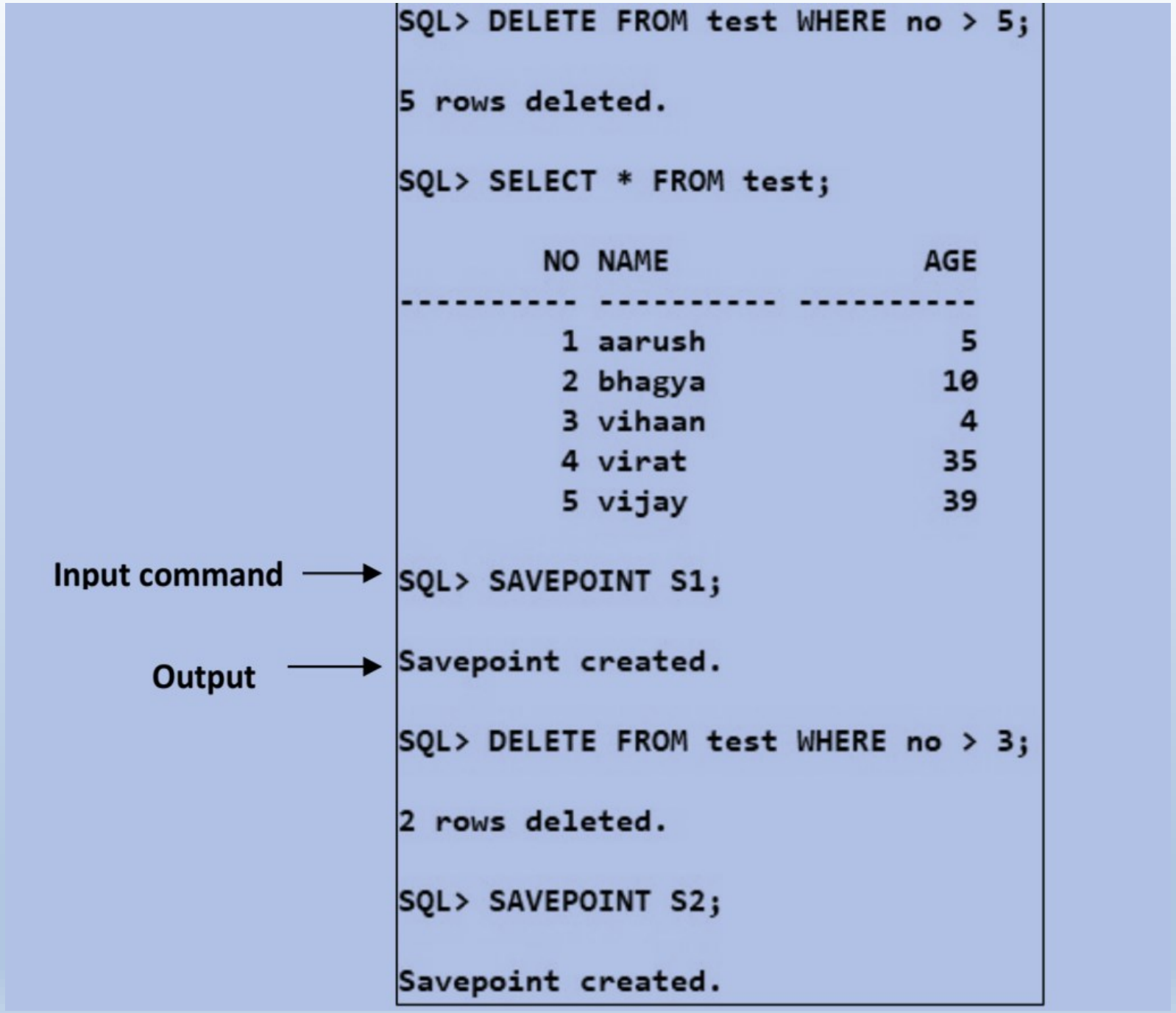
Savepoint Command

- The Savepoint command is used to identify a point in a transaction to which work can be undone
- It marks and save the current point in the processing of a transaction.
- The basic syntax is,

```
SAVEPOINT Savepoint_Name;
```

- Rollback and Savepoint work together to Rollback at a particular Savepoint that has been created earlier.

```
ROLLBACK TO SAVEPOINT savepoint_name;
```


SQL> DELETE FROM test WHERE no > 5;
5 rows deleted.
SQL> SELECT * FROM test;
NO NAME AGE
1 aarush 5
2 bhagya 10
3 vihaan 4
4 virat 35
5 vijay 39
Input command
SQL> SAVEPOINT S1;
Output
Savepoint created.
SQL> DELETE FROM test WHERE no > 3;
2 rows deleted.
SQL> SAVEPOINT S2;
Savepoint created.

```
SQL> ROLLBACK TO S1;

Rollback complete.

SQL> SELECT * FROM test;

        NO NAME                 AGE
---------- ---------- ----------
         1 aarush                5
         2 bhagya               10
         3 vihaan                4
         4 virat                35
         5 vijay                39
```

Distinct Keyword

- It is used to return only distinct (different/unique) values.
- Duplicates are eliminated.
- It is used with SELECT command.
- The syntax is,

SELECT DISTINCT columnName FROM tablename ;

Example

Display distinct (unique) id from table 'test'.

```
SQL> SELECT * FROM test;

        ID NAME                          AGE
---------- --------------- ----------
         1 bhagya                        15
         2 aarush                        10
         3 shreeja                        7
         4 kahaan                         3
         2 aarav                          5
         4 avani                         10

6 rows selected.

SQL> SELECT DISTINCT id FROM test;

        ID
----------
         1
         2
         4
         3
```

Display distinct (unique) age from table 'test'.

```
SQL> SELECT * FROM test;

        ID NAME                          AGE
---------- --------------- ----------
         1 bhagya                        15
         2 aarush                        10
         3 shreeja                        7
         4 kahaan                         3
         2 aarav                          5
         4 avani                         10

6 rows selected.

SQL> SELECT DISTINCT age FROM test;

       AGE
----------
         5
         7
         3
        15
        10
```

Copy Data From One Table to Another

- The syntax is,

```
CREATE TABLE newTableName (column1, column2,..,, columnN)
AS SELECT column1, column2,…,columnN
FROM SourceTableName
WHERE condition;
```

```
CRAETE TABLE newTableName
AS SELECT * FROM SourceTable;
```

Example

```
SQL> CREATE TABLE test1
  2  AS SELECT * FROM test;

Table created.

SQL> SELECT * FROM test1;

        ID NAME                 AGE
---------- -------------- ----------
         1 bhagya                15
         2 aarush                10
         3 shreeja                7
         4 kahaan                 3
         2 aarav                  5
         4 avani                 10

6 rows selected.
```

```
SQL> CREATE TABLE test2 (id, name)
  2  AS SELECT id, name FROM test;

Table created.

SQL> SELECT * FROM test2;

        ID NAME
---------- --------------
         1 bhagya
         2 aarush
         3 shreeja
         4 kahaan
         2 aarav
         4 avani

6 rows selected.
```

Create New User

- When we want to create a new user, we first have to login with our regular admin user (here we usually login into SYSTEM user), then using CREATE USER command we can create a new user.
- The syntax is,

```
CREATE USER user_Name
IDENTIFIED BY password;
```

```
SQL> connect system
Enter password:
Connected.
SQL> CREATE USER UNP
     IDENTIFIED BY uresh;

User created.
```

Grant Command

- The GRANT command is used to granting privileges means to give permission to some user to access database object or a part of a database object.
- The syntax is,

```
GRANT  object_privileges
ON object_name
TO user_name
WITH GRANT OPTION;
```

- The Owner of the database can grant all or some specific privileges to other users.
- The WITH GRANT OPTION allows the grantee, user to which privilege is granted to in turn grant object privilege to other users.

- The table, given in below illustrates various object privileges.

Privilege	Allows user
ALL	To perform all the operation listed below.
ALTER	To change the table structure using ALTER command.
DELETE	To delete records from the table using DELETE command
INDEX	To create an index on the table using CREATE INDEX command
INSERT	To insert records into the table using INSERT INTO command.
REFERENCES	To reference table while creating foreign keys.
SELECT	To query the table using SELECT command.
UPDATE	To modify the records in the table using UPDATE command.

Example

Suppose we have two users → system and UNP. We have a table 'test' which is created by the user SYSTEM (so we can say test table is owned by the user whose name is SYSTEM). Here by login in 'SYSTEM' user, we will grant SELECT permission to UNP. Before granting the permission UNP user cannot access 'test' table, which we can see in below figure.

```
SQL> CONNECT UNP
Enter password:
Connected.
SQL> SELECT * FROM system.test;
SELECT * FROM system.test
                    *
ERROR at line 1:
ORA-00942: table or view does not exist
```

Now we are giving SELECT permission to the user UNP (to give permission, SYSTEM user must be logged in).

Input command →

```
SQL> GRANT SELECT
     ON test
     TO UNP;
```

Output →

```
Grant succeeded.
```

Now, the user UNP is able to view the table 'test' of SYSTEM user.

Input command →

Output →

```
SQL> CONNECT UNP
Enter password:
Connected.
SQL> SELECT * FROM system.test;

        NO NAME                 AGE
---------- ---------- ----------
         1 aarush                5
         2 bhagya               10
         3 vihaan                4
         4 virat                35
         5 vijay                39
         6 sachin               25
         7 avani                15
         8 mahima               23
         9 shreeja               2
        10 aarav                 6

10 rows selected.
```

To access other user's table, dot operator is user along with user name.

Example

Input command →

Output →

```
SQL> GRANT SELECT
     ON test
     TO UNP;

Grant succeeded.
```

Here, we have logged in from system user and grant the permission of SELECT for *test* table to UNP user. But we have not given WITH GRANT OPTION. So the user UNP cannot further grant the privileges of *test* table. That we can see in below figure.

```
SQL> GRANT SELECT
     ON system.test
     TO KGP;
ON system.test
          *
ERROR at line 2:
ORA-01031: insufficient privileges
```

Now, if we provide WITH GRANT OPTION to UNP, then he further can grant privileges to other user. Check in below figure.

```
SQL> GRANT SELECT
     ON test
     TO UNP
     WITH GRANT OPTION;

Grant succeeded.
```

[system user logged in]

```
SQL> GRANT SELECT
     ON system.test
     TO KGP;

Grant succeeded.
```

[UNP user logged in]

Now, by logging in to user KGP, we can use SELECT command on *test* table which is owned by the system user.

Revoke Command

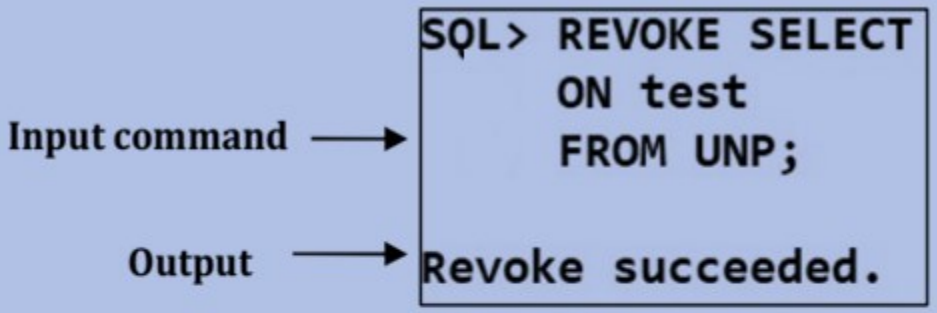
- The REVOKE command is used to withdraw access privileges given to users on the database.

- The owner on an object can revoke privileges granted to another user. A user of the object, who is not an owner, but has been granted privileges using WITH GRANT OPTION, can revoke the privilege from the grantee.

- The syntax is,

```
REVOKE  object_privileges
ON object_name
FROM user_name;
```

Example



[Logged in system user]

Here in above figure, we have revoked the SELECT permission from UNP user. Now UNP cannot use SELECT command on *test* table. That we can see below.

```
SQL> connect UNP
Enter password:
Connected.
SQL> SELECT * FROM system.test;
SELECT * FROM system.test
                     *
ERROR at line 1:
ORA-00942: table or view does not exist
```

GRANT	REVOKE
– Grant command is specifically used to provide permission to database objects for a user.	– Revoke command is used to remove the permissions or privileges of a user on database objects
– Syntax: GRANT permission_name ON object_name TO user_name [WITH GRANT OPTION];	– Syntax: REVOKE permission_name ON object_name FROM user_name;
– Grant gives access rights to the users.	– Revoke removes the access rights of all users
– When the access is decentralized granting permissions will be easy.	– If decentralized access removing the granted permissions is difficult.