UNIT 2-RELATIONAL MODEL

Relational Data Model - keys, referential integrity and foreign keys, Relational Algebra - SQL fundamentals- Introduction, data definition in SQL, table, key and foreign key definitions, update behaviors-Intermediate SQL-Advanced SQL features -Embedded SQL- Dynamic SQL, CASE Studies- Oracle:Database Design and Querying Tools; SQL Variations and Extensions

RELATIONAL DATA MODEL

Relational model can represent as a table with columns and rows. Each row is known as a tuple. Each table of the column has a name or attribute.

Domain: It contains a set of atomic values that an attribute can take.

Attribute: It contains the name of a column in a particular table. Each attribute Ai must have a domain, dom(Ai)

Relational instance: In the relational database system, the relational instance is represented by a finite set of tuples. Relation instances do not have duplicate tuples.

Relational schema: A relational schema contains the name of the relation and name of all columns or attributes.

Relational key: In the relational key, each row has one or more attributes. It can identify the row in the relation uniquely.

Example: STUDENT Relation

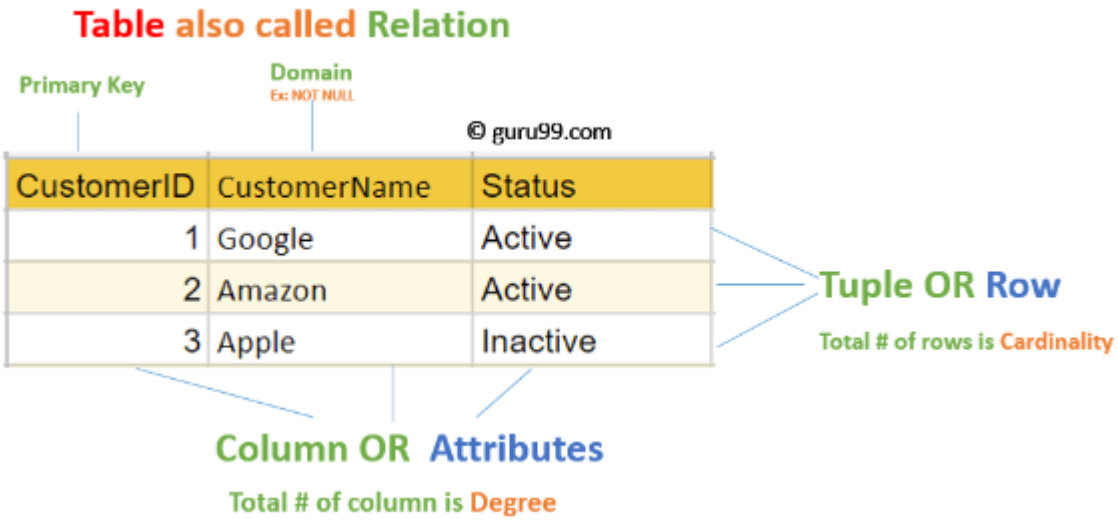
NAME	ROLL_NO	PHONE_NO	ADDRESS	AGE
Ram	14795	7305758992	Noida	24
Shyam	12839	9026288936	Delhi	35
Laxman	33289	8583287182	Gurugram	20
Mahesh	27857	7086819134	Ghaziabad	27

Ganesh	17282	9028 9i3988	Delhi	40

- In the given table, NAME, ROLL_NO, PHONE_NO, ADDRESS, and AGE are the attributes.
- The instance of schema STUDENT has 5 tuples.
- t3 = <Laxman, 33289, 8583287182, Gurugram, 20>

Properties of Relations

- Name of the relation is distinct from all other relations.
- Each relation cell contains exactly one atomic (single) value
- Each attribute contains a distinct name
- Attribute domain has no significance
- tuple has no duplicate value
- Order of tuple can have a different sequence



RELATIONAL INTEGRITY CONSTRAINTS

Relational Integrity constraints in DBMS are referred to conditions which must be present for a valid relation. These Relational constraints in DBMS are derived from the rules in the mini-world that the database represents.

There are many types of Integrity Constraints in DBMS. Constraints on the Relational database management system is mostly divided into three main categories are:

1. Domain Constraints
2. Key Constraints
3. Referential Integrity Constraints

DOMAIN CONSTRAINTS

Domain constraints can be violated if an attribute value is not appearing in the corresponding domain or it is not of the appropriate data type.

Domain constraints specify that within each tuple, and the value of each attribute must be unique. This is specified as data types which include standard data types integers, real numbers, characters, Booleans, variable length strings, etc.

Example:

Create DOMAIN CustomerName
CHECK (value not NULL)
The example shown demonstrates creating a domain constraint such that CustomerName is not NULL

KEY CONSTRAINTS

An attribute that can uniquely identify a tuple in a relation is called the key of the table. The value of the attribute for different tuples in the relation has to be unique.

Example:

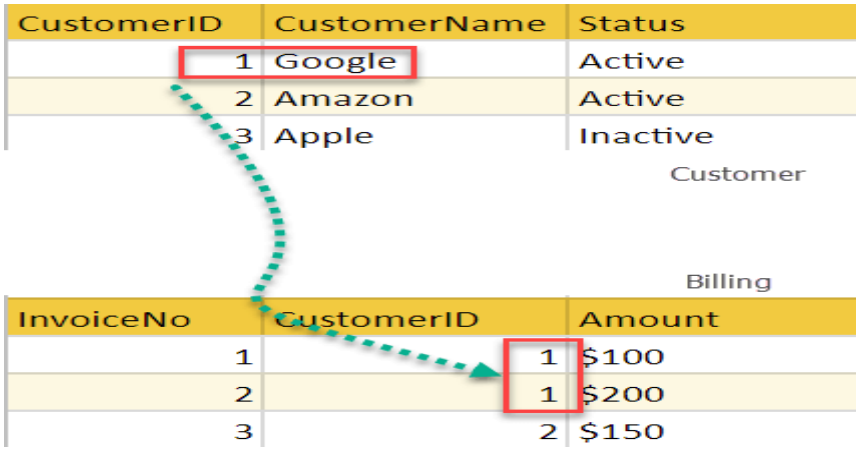
In the given table, CustomerID is a key attribute of Customer Table. It is most likely to have a single key for one customer, CustomerID =1 is only for the CustomerName =" Google".

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive

Referential Integrity Constraints

Referential Integrity constraints in DBMS are based on the concept of Foreign Keys. A foreign key is an important attribute of a relation which should be referred to in other relationships. Referential integrity constraint state happens where relation refers to a key attribute of a different or same relation. However, that key element must exist in the table.

Example:



In the above example, we have 2 relations, Customer and Billing.

Tuple for CustomerID =1 is referenced twice in the relation Billing. So we know CustomerName=Google has billing amount $300

OPERATIONS IN RELATIONAL MODEL

Four basic update operations performed on relational database model are

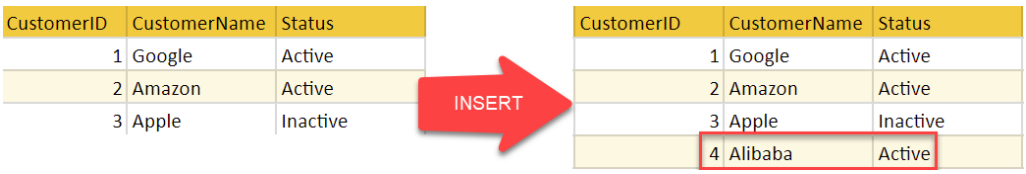
Insert, update, delete and select.

- Insert is used to insert data into the relation
- Delete is used to delete tuples from the table.
- Modify allows you to change the values of some attributes in existing tuples.
- Select allows you to choose a specific range of data.

Whenever one of these operations are applied, integrity constraints specified on the relational database schema must never be violated.

INSERT OPERATION

- The insert operation gives values of the attribute for a new tuple which should be inserted into a relation.



UPDATE OPERATION

- You can see that in the below-given relation table CustomerName= ‘Apple’ is updated from Inactive to Active.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Inactive
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

DELETE OPERATION

- To specify deletion, a condition on the attributes of the relation selects the tuple to be deleted.

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
3	Apple	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

- In the above-given example, CustomerName= “Apple” is deleted from the table.
- The Delete operation could violate referential integrity if the tuple which is deleted is referenced by foreign keys from other tuples in the same database.

SELECT OPERATION

CustomerID	CustomerName	Status
1	Google	Active
2	Amazon	Active
4	Alibaba	Active

CustomerID	CustomerName	Status
2	Amazon	Active

In the above-given example, CustomerName=”Amazon” is selected.

SQL

- SQL stands for Structured Query Language. It is used for storing and managing data in relational database management system (RDMS).
- It is a standard language for Relational Database System. It enables a user to create, read, update and delete relational databases and tables.
- All the RDBMS like MySQL, Informix, Oracle, MS Access and SQL Server use SQL as their standard database language.
- SQL allows users to query the database in a number of ways, using English-like statements.

Rules:

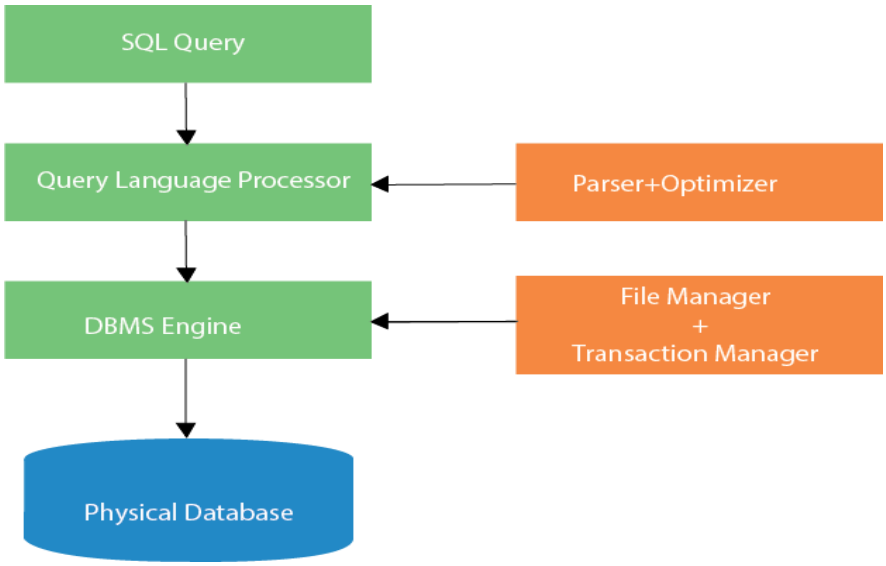
SQL follows the following rules:

- Structure query language is not case sensitive. Generally, keywords of SQL are written in uppercase.

- Statements of SQL are dependent on text lines. We can use a single SQL statement on one or multiple text line.
- Using the SQL statements, you can perform most of the actions in a database.
- SQL depends on tuple relational calculus and relational algebra.

SQL process:

- When an SQL command is executing for any RDBMS, then the system figure out the best way to carry out the request and the SQL engine determines that how to interpret the task.
- In the process, various components are included. These components can be optimization Engine, Query engine, Query dispatcher, classic, etc.
- All the non-SQL queries are handled by the classic query engine, but SQL query engine won't handle logical files.



CHARACTERISTICS OF SQL

- SQL is easy to learn.
- SQL is used to access data from relational database management systems.
- SQL can execute queries against the database.
- SQL is used to describe the data.
- SQL is used to define the data in the database and manipulate it when needed.
- SQL is used to create and drop the database and table.
- SQL is used to create a view, stored procedure, function in a database.

- SQL allows users to set permissions on tables, procedures, and views.

Advantages of SQL

There are the following advantages of SQL:

High speed

Using the SQL queries, the user can quickly and efficiently retrieve a large amount of records from a database.

No coding needed

In the standard SQL, it is very easy to manage the database system. It doesn't require a substantial amount of code to manage the database system.

Well defined standards

Long established are used by the SQL databases that are being used by ISO and ANSI.

Portability

SQL can be used in laptop, PCs, server and even some mobile phones.

Interactive language

SQL is a domain language used to communicate with the database. It is also used to receive answers to the complex questions in seconds.

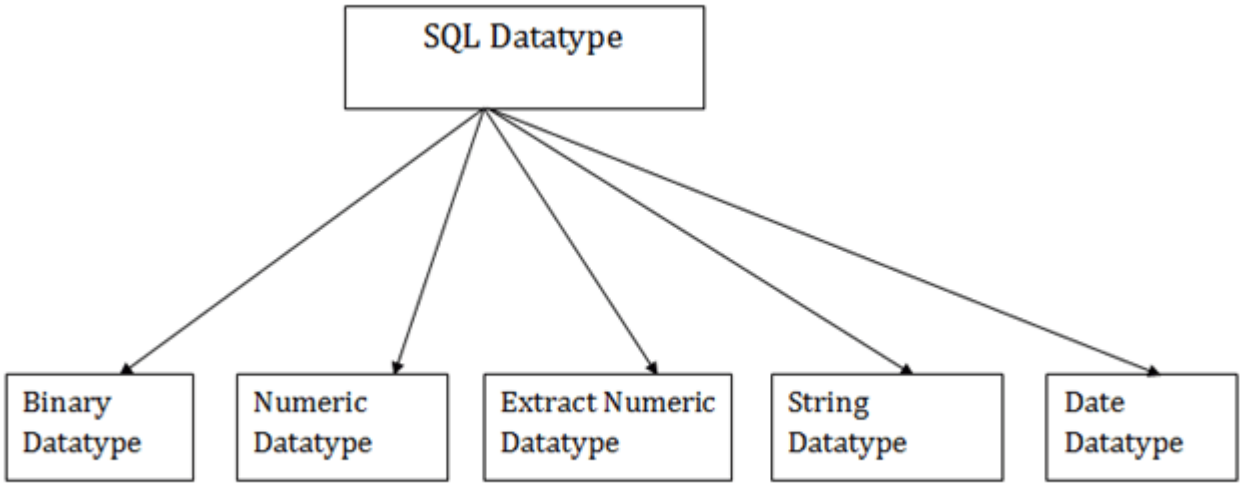
Multiple data view

Using the SQL language, the users can make different views of the database structure.

SQL Datatype

- SQL Datatype is used to define the values that a column can contain.
- Every column is required to have a name and data type in the database table.

Datatype of SQL:



1. Binary Datatypes

There are Three types of binary Datatypes which are given below:

Data Type	Description
binary	It has a maximum length of 8000 bytes. It contains fixed-length binary data.
varbinary	It has a maximum length of 8000 bytes. It contains variable-length binary data.
image	It has a maximum length of 2,147,483,647 bytes. It contains variable-length binary data.

2. Approximate Numeric Datatype :

The subtypes are given below:

Data type	From	To	Description

float	-1.79E + 308	1.79E + 308	It is used to specify a floating-point value e.g. 6.2, 2.9 etc.
real	-3.40e + 38	3.40E + 38	It specifies a single precision floating point number

3. Exact Numeric Datatype

The subtypes are given below:

Data type	Description
int	It is used to specify an integer value.
smallint	It is used to specify small integer value.
bit	It has the number of bits to store.
decimal	It specifies a numeric value that can have a decimal number.
numeric	It is used to specify a numeric value.

4. Character String Datatype

The subtypes are given below:

Data type	Description
char	It has a maximum length of 8000 characters. It contains Fixed-length non-unicode characters.

varchar	It has a maximum length of 8000 characters. It contains variable-length non-unicode characters.
text	It has a maximum length of 2,147,483,647 characters. It contains variable-length non-unicode characters.

5. Date and time Datatypes

The subtypes are given below:

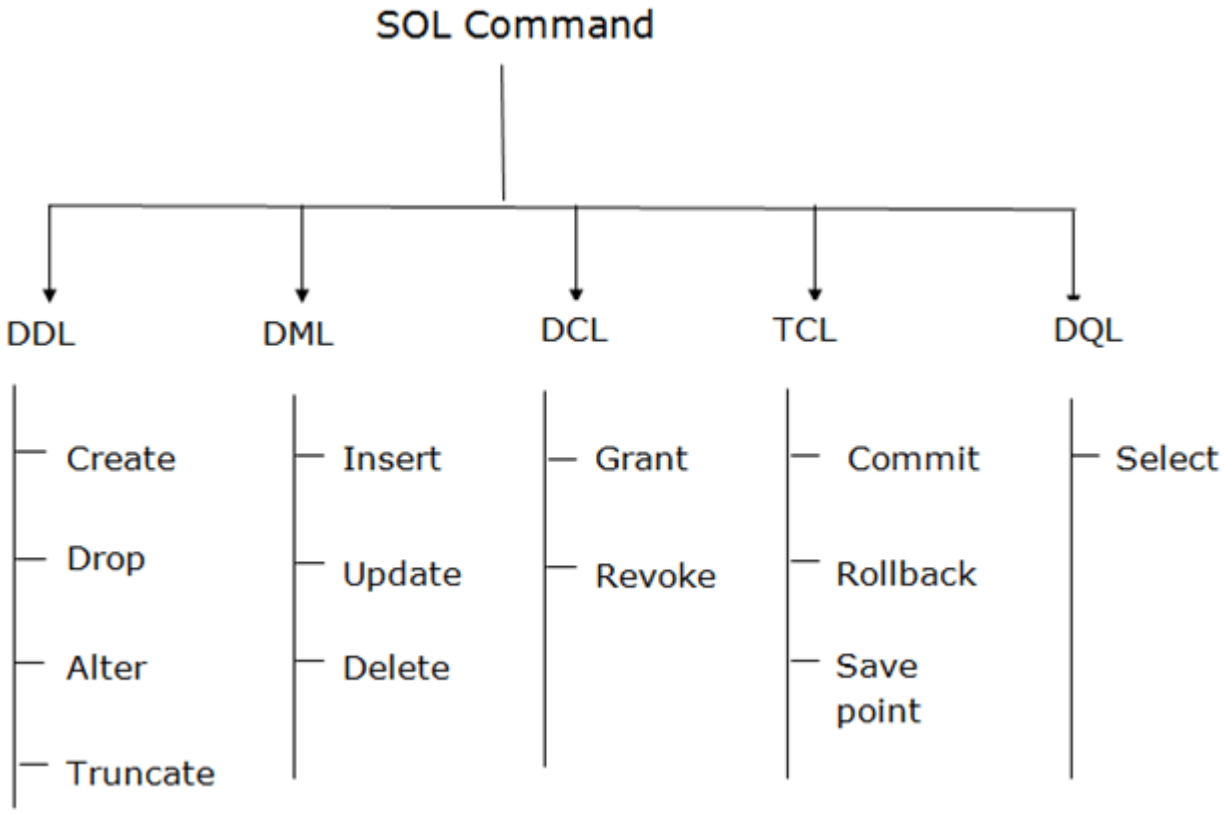
Datatype	Description
date	It is used to store the year, month, and days value.
time	It is used to store the hour, minute, and second values.
timestamp	It stores the year, month, day, hour, minute, and the second value.

SQL Commands

- SQL commands are instructions. It is used to communicate with the database. It is also used to perform specific tasks, functions, and queries of data.
- SQL can perform various tasks like create a table, add data to tables, drop the table, modify the table, set permission for users.

Types of SQL Commands

There are five types of SQL commands: DDL, DML, DCL, TCL, and DQL.



1. Data Definition Language (DDL)

- DDL changes the structure of the table like creating a table, deleting a table, altering a table, etc.
- All the command of DDL are auto-committed that means it permanently save all the changes in the database.

Here are some commands that come under DDL:

- CREATE
- ALTER
- DROP
- TRUNCATE

a. CREATE It is used to create a new table in the database.

Syntax:

CREATE TABLE TABLE_NAME (COLUMN_NAME DATATYPES[,....]);

Example:

1. CREATE TABLE EMPLOYEE(Name VARCHAR2(20), Email VARCHAR2(100), DOB DATE);

b. DROP: It is used to delete both the structure and record stored in the table.

Syntax

1. DROP TABLE table_name;

Example

1. DROP TABLE EMPLOYEE;

c. ALTER: It is used to alter the structure of the database. This change could be either to modify the characteristics of an existing attribute or probably to add a new attribute.

Syntax:

To add a new column in the table

1. ALTER TABLE table_name ADD column_name COLUMN-definition;

To modify existing column in the table:

1. ALTER TABLE table_name MODIFY(column_definitions....);

EXAMPLE

1. ALTER TABLE STU_DETAILS ADD(ADDRESS VARCHAR2(20));
2. ALTER TABLE STU_DETAILS MODIFY (NAME VARCHAR2(20));

d. TRUNCATE: It is used to delete all the rows from the table and free the space containing the table.

Syntax:

1. TRUNCATE TABLE table_name;

Example:

1. TRUNCATE TABLE EMPLOYEE;

2. Data Manipulation Language

- DML commands are used to modify the database. It is responsible for all form of changes in the database.

- The command of DML is not auto-committed that means it can't permanently save all the changes in the database. They can be rollback.

Here are some commands that come under DML:

- INSERT
- UPDATE
- DELETE

a. INSERT: The INSERT statement is a SQL query. It is used to insert data into the row of a table.

Syntax:

```
1. INSERT INTO TABLE_NAME
2. (col1, col2, col3,.... col N)
3. VALUES (value1, value2, value3, .... valueN);
```

Or

INSERT INTO TABLE_NAME VALUES (value1, value2, value3, ... valueN);

For example:

INSERT INTO javatpoint (Author, Subject) VALUES ("Sonoo", "DBMS");

b. UPDATE: This command is used to update or modify the value of a column in the table.

Syntax:

UPDATE table_name SET [column_name1= value1,...column_nameN = valueN] [WHERE CONDITION]

For example:

UPDATE students SET User_Name = 'Sonoo' WHERE Student_Id = '3'

c. DELETE: It is used to remove one or more row from a table.

Syntax:

DELETE FROM table_name [WHERE condition];

For example:

```
DELETE FROM javatpoint  WHERE Author="Sonoo";
```

3. Data Control Language

DCL commands are used to grant and take back authority from any database user.

Here are some commands that come under DCL:

- Grant
- Revoke

a. Grant: It is used to give user access privileges to a database.

Example

```
GRANT SELECT, UPDATE ON MY_TABLE TO SOME_USER, ANOTHER_USER;
```

b. Revoke: It is used to take back permissions from the user.

Example

```
REVOKE SELECT, UPDATE ON MY_TABLE FROM USER1, USER2;
```

4. Transaction Control Language

TCL commands can only use with DML commands like INSERT, DELETE and UPDATE only.

These operations are automatically committed in the database that's why they cannot be used while creating tables or dropping them.

Here are some commands that come under TCL:

- COMMIT
- ROLLBACK
- SAVEPOINT

a. Commit: Commit command is used to save all the transactions to the database.

Syntax:

```
COMMIT;
```

Example:

```
DELETE FROM CUSTOMERS  WHERE AGE = 25;
```

COMMIT;

b. Rollback: Rollback command is used to undo transactions that have not already been saved to the database.

Syntax:

ROLLBACK;

Example:

DELETE FROM CUSTOMERS WHERE AGE = 25;
ROLLBACK;

c. SAVEPOINT: It is used to roll the transaction back to a certain point without rolling back the entire transaction.

Syntax:

SAVEPOINT SAVEPOINT_NAME;

5. Data Query Language

DQL is used to fetch the data from the database.

It uses only one command:

- SELECT

a. SELECT: This is the same as the projection operation of relational algebra. It is used to select the attribute based on the condition described by WHERE clause.

Syntax:

SELECT expressions FROM TABLES WHERE conditions;

For example:

SELECT emp_name FROM employee WHERE age > 20;

SQL Table

- SQL Table is a collection of data which is organized in terms of rows and columns. In DBMS, the table is known as relation and row as a tuple.

- Table is a simple form of data storage. A table is also considered as a convenient representation of relations.

Let's see an example of the **EMPLOYEE** table:

EMP_ID	EMP_NAME	CITY	PHONE_NO
1	Kristen	Washington	7289201223
2	Anna	Franklin	9378282882
3	Jackson	Bristol	9264783838
4	Kellan	California	7254728346
5	Ashley	Hawaii	9638482678

In the above table, "EMPLOYEE" is the table name, "EMP_ID", "EMP_NAME", "CITY", "PHONE_NO" are the column names. The combination of data of multiple columns forms a row, e.g., 1, "Kristen", "Washington" and 7289201223 are the data of one row.

Operation on Table

1. Create table
2. Drop table
3. Delete table
4. Rename table

SQL Create Table

SQL create table is used to create a table in the database. To define the table, you should define the name of the table and also define its columns and column's data type.

Syntax

```
create table "table_name"
("column1" "data type",
"column2" "data type",
"column3" "data type",
```

...
"columnN" "data type");

Example

```
SQL> CREATE TABLE EMPLOYEE (
EMP_ID INT                    NOT NULL,
EMP_NAME VARCHAR (25) NOT NULL,
PHONE_NO INT                  NOT NULL,
ADDRESS CHAR (30),
PRIMARY KEY (ID)
);
```

If you create the table successfully, you can verify the table by looking at the message by the SQL server. Else you can use DESC command as follows:

SQL> DESC EMPLOYEE;

Field	Type	Null	Key	Default	Extra
EMP_ID	int(11)	NO	PRI	NULL	
EMP_NAME	varchar(25)	NO		NULL	
PHONE_NO	NO	int(11)		NULL	
ADDRESS	YES			NULL	char(30)

- 4 rows in set (0.35 sec)

Now you have an EMPLOYEE table in the database, and you can use the stored information related to the employees.

Drop table

A SQL drop table is used to delete a table definition and all the data from a table. When this command is executed, all the information available in the table is lost forever, so you have to very careful while using this command.

Syntax

DROP TABLE "table_name";

Firstly, you need to verify the **EMPLOYEE** table using the following command:

SQL> DESC EMPLOYEE;

Field	Type	Null	Key	Default	Extra
EMP_ID	int(11)	NO	PRI	NULL	
EMP_NAME	varchar(25)	NO		NULL	
PHONE_NO	NO	int(11)		NULL	
ADDRESS	YES			NULL	char(30)

- 4 rows in set (0.35 sec)

This table shows that EMPLOYEE table is available in the database, so we can drop it as follows:

SQL>DROP TABLE EMPLOYEE;

Now, we can check whether the table exists or not using the following command:

Query OK, 0 rows affected (0.01 sec)

As this shows that the table is dropped, so it doesn't display it.

SQL DELETE table

In SQL, DELETE statement is used to delete rows from a table. We can use WHERE condition to delete a specific row from a table. If you want to delete all the records from the table, then you don't need to use the WHERE clause.

Syntax

DELETE FROM table_name WHERE condition;

Example

Suppose, the EMPLOYEE table having the following records:

EMP_ID	EMP_NAME	CITY	PHONE_NO	SALARY
1	Kristen	Chicago	9737287378	150000
2	Russell	Austin	9262738271	200000
3	Denzel	Boston	7353662627	100000
4	Angelina	Denver	9232673822	600000
5	Robert	Washington	9367238263	350000
6	Christian	Los angels	7253847382	260000

The following query will DELETE an employee whose ID is 2.

```
SQL> DELETE FROM EMPLOYEE  WHERE EMP_ID = 3;
```

Now, the EMPLOYEE table would have the following records.

EMP_ID	EMP_NAME	CITY	PHONE_NO	SALARY
1	Kristen	Chicago	9737287378	150000
2	Russell	Austin	9262738271	200000
4	Angelina	Denver	9232673822	600000
5	Robert	Washington	9367238263	350000
6	Christian	Los angels	7253847382	260000

If you don't specify the WHERE condition, it will remove all the rows from the table.

```
DELETE FROM EMPLOYEE;
```

Now, the EMPLOYEE table would not have any records.

SQL SELECT Statement

In SQL, the SELECT statement is used to query or retrieve data from a table in the database. The returns data is stored in a table, and the result table is known as result-set.

Syntax

SELECT column1, column2, ... FROM table_name;

Here, the expression is the field name of the table that you want to select data from.

Use the following syntax to select all the fields available in the table:

SELECT * FROM table_name;

Example:

EMPLOYEE

EMP_ID	EMP_NAME	CITY	PHONE_NO	SALARY
1	Kristen	Chicago	9737287378	150000
2	Russell	Austin	9262738271	200000
3	Angelina	Denver	9232673822	600000
4	Robert	Washington	9367238263	350000
5	Christian	Los angels	7253847382	260000

To fetch the EMP_ID of all the employees, use the following query:

SELECT EMP_ID FROM EMPLOYEE;

Output

EMP_ID
1

2
3
4
5

To fetch the EMP_NAME and SALARY, use the following query:

SELECT EMP_NAME, SALARY FROM EMPLOYEE;

EMP_NAME	SALARY
Kristen	150000
Russell	200000
Angelina	600000
Robert	350000
Christian	260000

To fetch all the fields from the EMPLOYEE table, use the following query:

SELECT * FROM EMPLOYEE

Output

EMP_ID	EMP_NAME	CITY	PHONE_NO	SALARY
1	Kristen	Chicago	9737287378	150000
2	Russell	Austin	9262738271	200000
3	Angelina	Denver	9232673822	600000

4	Robert	Washington	9367238263	350000
5	Christian	Los angels	7253847382	260000

SQL INSERT Statement

The SQL INSERT statement is used to insert a single or multiple data in a table. In SQL, You can insert the data in two ways:

1. Without specifying column name
2. By specifying column name

Sample Table

EMPLOYEE

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36

1. Without specifying column name

If you want to specify all column values, you can specify or ignore the column values.

Syntax

INSERT INTO TABLE_NAME VALUES (value1, value2, value 3, ... Value N);

Query

INSERT INTO EMPLOYEE VALUES (6, 'Marry', 'Canada', 600000, 48);

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

2. By specifying column name

To insert partial column values, you must have to specify the column names.

Syntax

INSERT INTO TABLE_NAME [(col1, col2, col3,.... col N)] VALUES (value1, value2, value 3, ... Value N);

Query

INSERT INTO EMPLOYEE (EMP_ID, EMP_NAME, AGE) VALUES (7, 'Jack', 40);

Output: After executing this query, the table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29

5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48
7	Jack	null	null	40

SQL Update Statement

The SQL UPDATE statement is used to modify the data that is already in the database. The condition in the WHERE clause decides that which row is to be updated.

Syntax

UPDATE table_name SET column1 = value1, column2 = value2, ... WHERE condition;

Sample Table

EMPLOYEE

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Updating single record

Update the column EMP_NAME and set the value to 'Emma' in the row where SALARY is 500000.

Syntax

UPDATE table_name SET column_name = value WHERE condition;

Query

```
UPDATE EMPLOYEE   SET EMP_NAME = 'Emma'   WHERE SALARY = 500000;
```

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Emma	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Updating multiple records

If you want to update multiple columns, you should separate each field assigned with a comma. In the EMPLOYEE table, update the column EMP_NAME to 'Kevin' and CITY to 'Boston' where EMP_ID is 5.

Syntax

```
UPDATE table_name    SET column_name = value1, column_name2 = value2   WHERE co
ndition;
```

Query

```
UPDATE EMPLOYEE   SET EMP_NAME = 'Kevin', City = 'Boston'   WHERE EMP_ID = 5;
```

Output

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30

2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Kevin	Boston	200000	36
6	Marry	Canada	600000	48

Without use of WHERE clause

If you want to update all row from a table, then you don't need to use the WHERE clause. In the EMPLOYEE table, update the column EMP_NAME as 'Harry'.

Syntax

```
UPDATE table_name   SET column_name = value1;
```

Query

```
UPDATE EMPLOYEE   SET EMP_NAME = 'Harry';
```

Output

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Harry	Chicago	200000	30
2	Harry	Austin	300000	26
3	Harry	Denver	100000	42
4	Harry	Washington	500000	29
5	Harry	Los angels	200000	36
6	Harry	Canada	600000	48

SQL DELETE Statement

The SQL DELETE statement is used to delete rows from a table. Generally, DELETE statement removes one or more records form a table.

Syntax

DELETE FROM table_name WHERE some_condition;

Sample Table

EMPLOYEE

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
4	Kristen	Washington	500000	29
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Deleting Single Record

Delete the row from the table EMPLOYEE where EMP_NAME = 'Kristen'. This will delete only the fourth row.

Query

DELETE FROM EMPLOYEE WHERE EMP_NAME = 'Kristen';

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
1	Angelina	Chicago	200000	30

2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Deleting Multiple Record

Delete the row from the EMPLOYEE table where AGE is 30. This will delete two rows(first and third row).

Query

1. DELETE FROM EMPLOYEE WHERE AGE= 30;

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE
2	Robert	Austin	300000	26
3	Christian	Denver	100000	42
5	Russell	Los angels	200000	36
6	Marry	Canada	600000	48

Delete all of the records

Delete all the row from the EMPLOYEE table. After this, no records left to display. The EMPLOYEE table will become empty.

Syntax

1. DELETE * FROM table_name;
2. or
3. DELETE FROM table_name;

Query

1. DELETE FROM EMPLOYEE;

Output: After executing this query, the EMPLOYEE table will look like:

EMP_ID	EMP_NAME	CITY	SALARY	AGE

Note: Using the condition in the WHERE clause, we can delete single as well as multiple records. If you want to delete all the records from the table, then you don't need to use the WHERE clause.

Views in SQL

- Views in SQL are considered as a virtual table. A view also contains rows and columns.
- To create the view, we can select the fields from one or more tables present in the database.
- A view can either have specific rows based on certain condition or all the rows of a table.

Advantages of View:

1. **Complexity:** Views help to reduce the complexity. Different views can be created on the same base table for different users.
2. **Security:** It increases the security by excluding the sensitive information from the view.
3. **Query Simplicity:** It helps to simplify commands from the user. A view can draw data from several different tables and present it as a single table.
4. **Consistency:** A view can present a consistent, unchanged image of the structure of the database. Views can be used to rename the columns without affecting the base table.
5. **Data Integrity:** If data is accessed and entered through a view, the DBMS can automatically check the data to ensure that it meets the specified integrity constraints.
6. **Storage Capacity:** Views take very little space to store the data.
7. **Logical Data Independence:** View can make the application and database tables to a certain extent independent.

Disadvantages of View:

The DML statements which can be performed on a view created using single base table have certain restrictions are:

1. You cannot INSERT if the base table has any not null column that do not appear in view.
2. You cannot INSERT or UPDATE if any of the column referenced in the INSERT or UPDATE contains group functions or columns defined by expression.
3. You can't execute INSERT, UPDATE, DELETE statements on a view if with read only option is enabled.
4. You can't be created view on temporary tables.
5. You cannot INSERT, UPDATE, DELETE if the view contains group functions GROUP BY, DISTINCT or a reference to a psuedocolumn rownum.
6. You can't pass parameters to the SQL server views.
7. You can't associate rules and defaults with views.

Sample table:

Student_Detail

STU_ID	NAME	ADDRESS
1	Stephan	Delhi
2	Kathrin	Noida
3	David	Ghaziabad
4	Alina	Gurugram

Student_Marks

STU_ID	NAME	MARKS	AGE
1	Stephan	97	19

2	Kathrin	86	21
3	David	74	18
4	Alina	90	20
5	John	96	18

1. Creating view

A view can be created using the **CREATE VIEW** statement. We can create a view from a single table or multiple tables.

Syntax:

CREATE VIEW view_name **AS SELECT** column1, column2..... **FROM** table_name **WHER** condition;

2. Creating View from a single table

In this example, we create a View named DetailsView from the table Student_Detail.

Query:

CREATE VIEW DetailsView **AS SELECT NAME**, ADDRESS **FROM** Student_Details **WHER E** STU_ID < 4;

Just like table query, we can query the view to view the data.

SELECT * **FROM** DetailsView;

Output:

NAME	ADDRESS
Stephan	Delhi
Kathrin	Noida
David	Ghaziabad

3. Creating View from multiple tables

View from multiple tables can be created by simply include multiple tables in the SELECT statement.

In the given example, a view is created named MarksView from two tables Student_Detail and Student_Marks.

Query:

CREATE VIEW MarksView **AS**
SELECT Student_Detail.**NAME**, Student_Detail.ADDRESS, Student_Marks.MARKS
FROM Student_Detail, Student_Mark
WHERE Student_Detail.**NAME** = Student_Marks.**NAME**;

To display data of View MarksView:

1. **SELECT** * **FROM** MarksView;

NAME	ADDRESS	MARKS
Stephan	Delhi	97
Kathrin	Noida	86
David	Ghaziabad	74
Alina	Gurugram	90

4. Deleting View

A view can be deleted using the Drop View statement.

Syntax

DROP VIEW view_name;

Example:

STATIC SQL

Static SQL refers to those SQL statements which are fixed and can be hard coded into the application. As static sqls are fixed queries, these statements can be analysed and optimized and do not require any specific handling for security purpose.

DYNAMIC SQL

Dynamic SQL refers to those SQL statements which are generated dynamically based on user's input and run in the application. Dynamic Sqls helps to develop general and flexible applications. Dynamic SQL may need more permissions and security handling and a malicious user can create dangerous code as well.

Following are some of the important differences between Static Routing and Dynamic Routing.

Sr. No.	Key	Static SQL	Dynamic SQL
1	Database Access	In Static SQL, database access procedure is predetermined in the statement.	In Dynamic SQL, how a database will be accessed, can be determine only at run time.
2	Efficiency	Static SQL statements are more faster and efficient.	Dynamic SQL statements are less efficient.
3	Compilation	Static SQL statements are compiled at compile time.	Dynamic SQL statements are compiled at run time.
4	Application Plan	Application Plan parsing, validation, optimization and generation are compile time activities.	Application Plan parsing, validation, optimization and generation are run time activities.
5	Use Cases	Static SQL is used in case of uniformly distributed data.	Dynamic SQL is used in case of non-uniformly distributed data.
6	Dynamic Statements	Statements like EXECUTE IMMEDIATE, EXECUTE, PREPARE are not used.	Statements like EXECUTE IMMEDIATE, EXECUTE, PREPARE are used
7	Flexibility	Static SQL is less flexible.	Dynamic SQL is highly flexible.

What is Oracle Database?

Oracle Database, or Oracle DB, is a relational database management system (RDBMS) developed by the Oracle corporation. It was originally developed by Lawrence Ellison together with other developers. Oracle DB is one of the most trusted and widely used database management systems across the world.

Oracle DB is based on a relational database framework where users can directly access database objects by running SQL queries. It is a scalable database management system, hence, it is used by global companies that store data across wide and local area networks. It also comes with its own network component to facilitate communication across networks.

Key Features of Oracle Database

Here a some of the key features responsible for the immense popularity of Oracle.

- **Cross-Platform Integration:** Oracle supports and works on all operating systems (OS), including Windows, macOS, Linux, and others.
- **Backup and Recovery:** Oracle's backup and recovery capabilities allow it to retrieve data from any accident or technical failure. Oracle's RAC architecture ensures that all data and processes are backed up.
- **Analytics Solutions:** You can use Oracle Advanced Analytics and OLAP (Oracle Analytic Processing) to quickly do analytical computations on business data.
- **Compliant with ACID Attributes:** Oracle DB offers ACID (Atomicity, Consistency, Isolation, and Durability) properties to ensure the Database's integrity during transaction processing.
- **Simple Communication:** Communication between applications on different platforms is simple. Oracle's native networking stack allows you to seamlessly interface your database with applications on a variety of platforms. For example, you can simply link and interact with a Unix-based application with your Oracle database (running on Windows).

Load your Oracle Data Using Hevo's No-Code Data Pipeline

Hevo Data, a Fully-managed No-Code Data Pipeline, can help you automate, simplify & enrich your data integration process in a few clicks. With Hevo's out-of-the-box connectors and blazing-fast Data Pipelines, you can extract data from 100+ Data Sources(**including 40+ free data sources**) for loading it straight into your Data Warehouse, Database, or any destination. To further streamline and prepare your data for analysis, you can process and enrich Raw Granular Data using Hevo's robust & built-in Transformation Layer without writing a single line of code!"

GET STARTED WITH HEVO FOR FREE

Hevo is the fastest, easiest, and most reliable data replication platform that will save your engineering bandwidth and time multifold. Try our 14-day full access free trial today to experience an entirely automated hassle-free Data Replication!

Accelerate your ETL with Hevo's Automated Data Platform. Try our 14-day full access free trial today!

Best Oracle Database Design Tool Options

Below are the top Oracle database design tool options:

1) Visual Paradigm ERD Tools

Image Source

Visual Paradigm is a top database design tool among database developers. It helps users to create a database design by following an Entity Relational Diagram (ERD) approach. An ERD is a baseline for any database. It shows how the tables within the database will be related to each other.

Visual Paradigm helps in making projects successful through features such as Enterprise Architecture, Mind Mapping, Visual Modeling, and more. You can use it to create various diagrams to represent your physical database.

The tool also generates DDL files to help you translate the physical model into a physical database.

2) ER Studio

Image Source

ER Studio is an oracle database design tool and data architecture tool created by IDERA, Inc. You can use it as a desktop app on your Microsoft computer. Other than Oracle, this top database design tool is compatible with cloud platforms such as Azure SQL Database, Amazon S3 & RDS, Google Database Service, Blob storage, Oracle MySQL Cloud Service, and Snowflake.

ER Studio comes with notations for drawing ER diagrams at all levels, that is, conceptual, logical, and physical. It also has an automated tool to help you translate your logical model into a physical model. ER Studio also checks for compliance and normalization.

3) Navicat

Image Source

Navicat is another top Oracle database design tool that can be installed on Windows, Linux, and Mac OS for designing databases. It has three standard notations, UML, Crow's Foot, and IDEF1X. It also comes with automated features for creating logical and physical models as well as physical databases.

This top database design tool has a reverse engineering feature that helps in importing existing supported databases to edit them visually. Its Export SQL feature allows users to generate SQL scripts for every component of the physical data model.

4) Lucidchart

Image Source

Lucidchart is another top Oracle database design tool. It comes with collaborative database design tools for creating database diagrams. This gives the teams proper assistance in creating a database that works properly.

Lucidchart is a web-based tool, hence, you don't have to download it to use it. Instead, you access it from your web browser. It is also a good collaboration tool, allowing database design teams to access their projects in real-time. Lucidchart comes with more than 500 image templates and supports different integrations other than Oracle.

What Makes Hevo's Data Loading Process Unique

Aggregating and Loading data Incrementally can be a mammoth task without the right set of tools. Hevo's automated platform empowers you with everything you need to have a smooth Data Collection, Processing, and Aggregation experience. Our platform has the following in store for you!

- **Built to Scale:** Exceptional Horizontal Scalability with Minimal Latency for Modern-data Needs.
- **Built-in Connectors:** Support for 100+ Data Sources, including Databases, SaaS Platforms, Files & More. Native Webhooks & REST API Connector available for Custom Sources.
- **Incremental Data Load**: Hevo allows the transfer of data that has been modified in real-time. This ensures efficient utilization of bandwidth on both ends.
- **Blazing-fast Setup:** Straightforward interface for new customers to work on, with minimal setup time.
- **Live Support**: The Hevo team is available round the clock to extend exceptional support to its customers through chat, email, and support calls.

SIGN UP HERE FOR A 14-DAY FREE TRIAL!

5) Vertabelo

Image Source

Vertabelo is an online Oracle database design tool that helps you to model your database from the conceptual model to the physical model. Other than Oracle, it supports several other database management systems including PostgreSQL, SQL Server, and others.

Vertabelo comes with a clean, responsive, and modern user interface. It also supports UML, Crow's Foot, and IDEF1X notations for creating logical and physical data models.

It also comes with an automated feature for creating the physical data model from the logical data model.

6) Erwin Data Modeler

Image Source

Erwin Data Modeler is also another top Oracle database design tool that can help you to deploy high-quality enterprise data assets. It helps organizations to discover, design, visualize, and deploy enterprise data through an intuitive user interface built on industry standards.

This database design tool also offers a model-driven collaboration and uses standards to ensure data quality. It is used by Data Architects and other IT professionals.

7) DbWrench

Image Source

DbWrench is an Oracle database design tool that supports synchronization and the designing of databases. It helps users save time when handling many tasks. With DbWrench, you can forward as well as reverse engineer databases.

DbWrench was developed by Nizar systems in 2017 with the goal of having an easy-to-use Oracle database design tool. You can also use it on different operating systems.

8) DeZign for Databases

Image Source

DeZign is another Oracle database design tool that supports data modeling for database professionals. It is a robust tool with easy-to-use features.

DeZign supports features such as ER diagramming, industry-standard design notations, multiple display modes, pan and zoom window, reverse engineering, ER modeling, and forward engineering.

It is a suitable tool for both beginner and experienced database designers.

Conclusion

The key pointers covered in this article are: Databases help organizations store data—the structure and group of data during storage. Oracle is a database management system. You can run SQL queries on your Oracle database to establish relations between different entities. Database design is a collection of processes to facilitate the designing, development, implementation, and maintenance of data management systems.

A proper database design helps to reduce the maintenance cost, improve data consistency and reduce disk storage costs. There are many Oracle database design tool options. It's up to you to choose the tool that fits the goals and requirements of your organization. In case you want to export data from a source of your choice such as **Oracle** into your desired Database/destination then **Hevo Data** is the right choice for you!