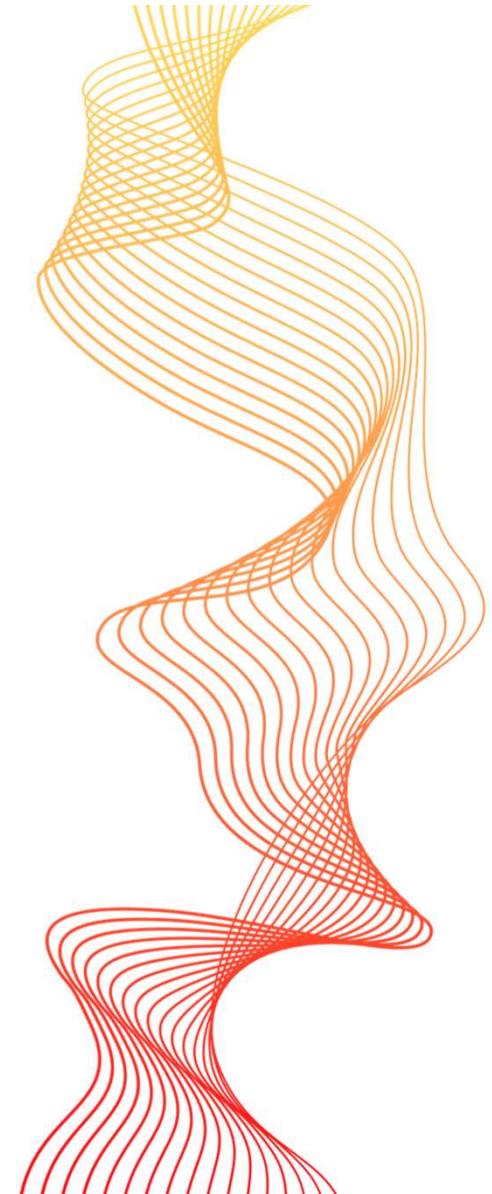


❖ *STRUCTURED
QUERY
LANGUAGE*



Database Types:

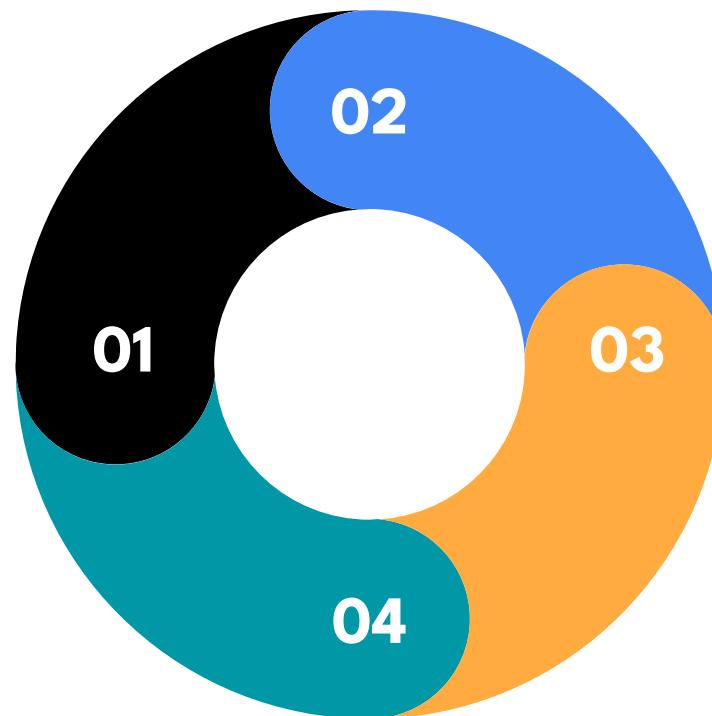
- ❖ This presentation provides an overview of SQL and different types of database management systems.
- **Relational Databases (RDBMS):**
Ex: MySQL, PostgreSQL, Oracle Database, Microsoft SQL server.
- **Description:** Relational databases organize data into structured tables with rows and columns. They use SQL (Structured Query Language) for data manipulation and querying. These databases are excellent for structured data with well-defined relationships.



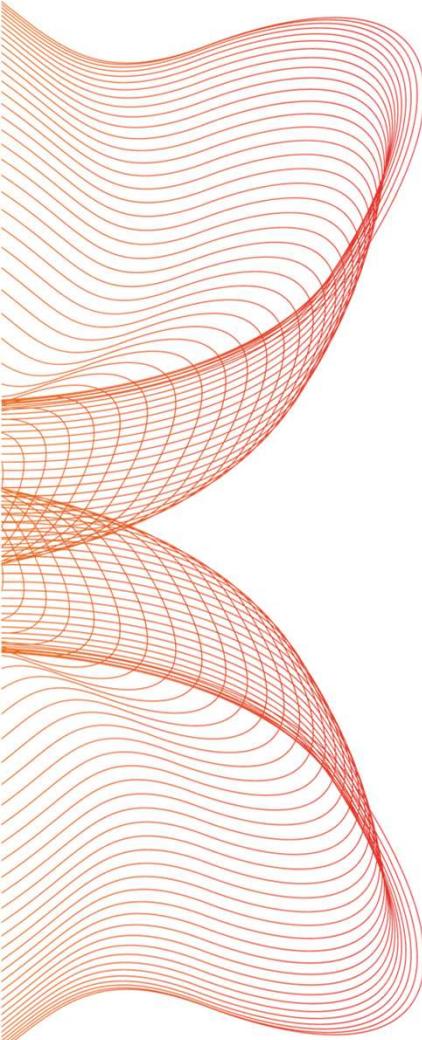
SQL Commands

SQL commands are categorized into three main groups:

- **DDL (Data Definition Language)**: Used for defining the database structure, including creating, altering, and dropping tables and schemas.



- **DML (Data Manipulation Language)**: Used for manipulating data within the tables, including SELECT, INSERT, UPDATE, and DELETE.
- ❖ **DCL (Data Control Language)**: Used for managing user and access, including GRANT and REVOKE.



CREATE STATEMENT:

- CREATE TABLE
table_name (column1_name
datatype constraint,
• column2_name datatype
constraint,
...columnN_name datatype
constraint);
- 

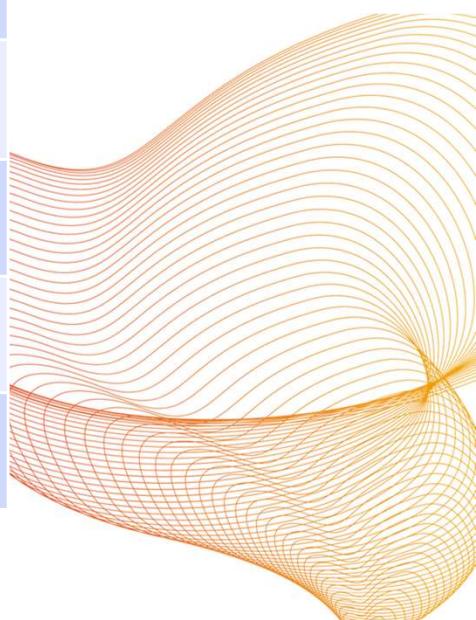


INSERT INTO STATEMENT:

- INSERT INTO table_name (column1, column2, column3, ...)VALUES (value1, value2, value3, ...);
- For better unnterstanding consider below Table:
- CREATE TABLE Customers (CustomerID INT PRIMARY KEY,FirstName VARCHAR(50),LastName VARCHAR(50),Age INT,City VARCHAR(50));
- INSERT INTO Customers (CustomerID, FirstName,LastName, Age, City) VALUES(1, 'John', 'Smith', 30, 'New York'),(2, 'Emily', 'Johnson', 25, 'New York'),(3, 'Michael', 'Williams', 40, 'Chicago'),(4, 'Emma', 'Brown', 28, 'Houston'),(5, 'William', 'Jones', 35, 'Miami');

 *CUSTOMERS TABLE:*

CustomerID	FirstName	LastName	Age	City
1	John	Smith	30	New York
2	Emily	Johnson	25	New York
3	Micheal	Williams	40	Chicago
4	Emma	Brown	28	Houston
5	William	Jones	35	Miami





SELECT STATEMENT:

- SELECT column1, column2, ...
- FROM table_name
- WHERE condition
- Select *(or)all tells considered duplicates(repeated),
 - ❖ select distinct consider (unique)
- **EX:**
- SELECT CUSTOMERID,FIRSTNAME
 FROM CUSTOMERS

ALTER STATEMENT

- ❖ ALTER TABLE ADD/DROP CONSTARINT COLUMN CONSTRAINT;

EX:ALTER CUSTOMERS ADD CONSTRAINT FIRSTNAME UNIQUE;



- ❖ ALTER TABLE ADD/DROP/MODIFY COLUMN DATATYPE;

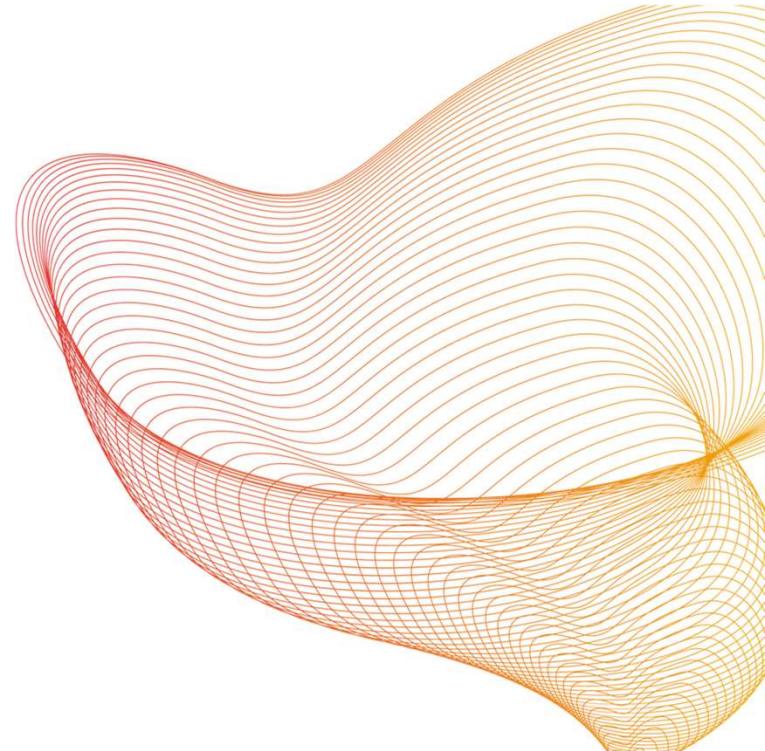
EX:ALTER CUSTOMERS ADD DISCOUNT FLOAT;

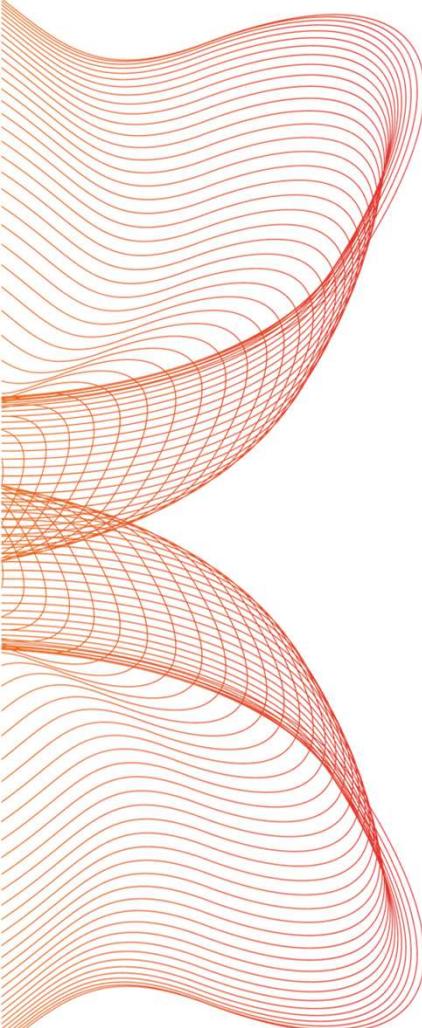
- ❖ FOR DROP DON'T NEED TO WRITE DATATYPE WHILE FOR REMAINING IT IS NECESSARY FOR BOTH ABOVE SYNTAXES.



WHERE CONDITION:

- SELECT column1, column2,
...
FROM table_name
WHERE condition;
- EX: SELECT * FROM
ORDERS WHERE
CUSTOMERID=1;
- This prints the row where
CustomerID is 1.





UPDATE STATEMENT (MODIFICATION):

01 **Syntax:** UPDATE table_name SET column1 = value1,
column2 = value2, ... WHERE condition

02 **Example:**
 update customers
set age=16
where customerid=2;



DELETE STATEMENT:

01 **Syntax:** DELETE FROM table_name WHERE condition;

02 Example: delete from customers where age=25

➤ **Note:** delete from table removes all the data;



EXISTS:

01

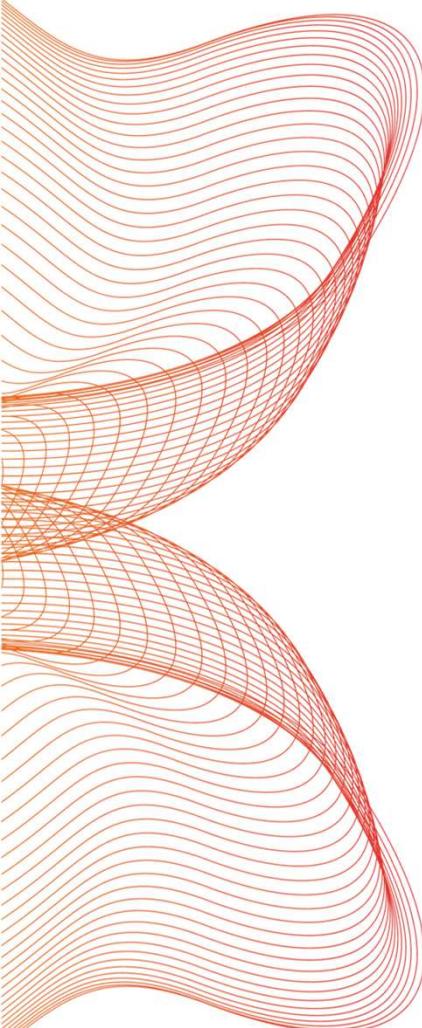
SYNTAX:

```
SELECT column1, column2, ...
FROM table_name
WHERE EXISTS (subquery);
```

02

```
EX:select age
From customers
Where customerid exists(select customerid
from customers where city='New York';
```





PRIMARY KEYS:

- A primary key is a unique identifier for each record in a table.
 - It ensures data integrity and links tables together.
 - **Syntax:** ATTRIBUTE DATATYPE
PRIMARY KEY;
 - The primary key must contain a unique value for each record in the table. No two records in the same table can have the same value(s) in the primary key field(s). This uniqueness ensures that each record can be uniquely identified.
- 



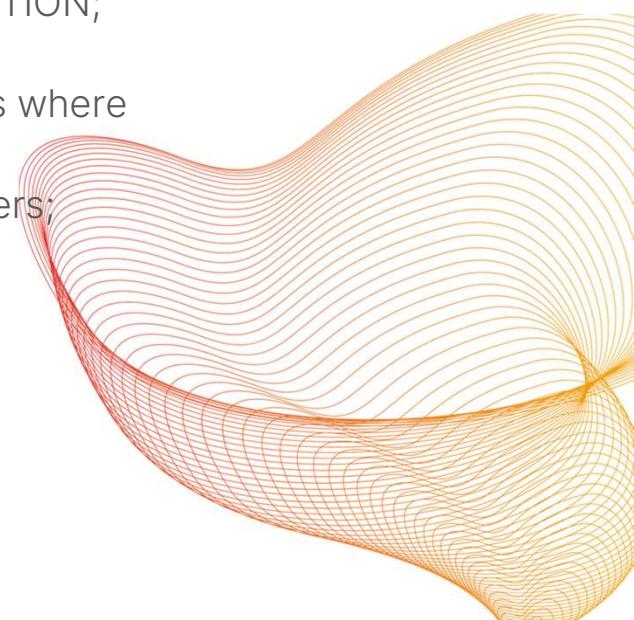
FOREIGN KEYS:

- ❖ A foreign key is a column in one table that refers to the primary key in another table.
- ❖ It establishes relationships between tables and enforces referential integrity constraints.
- ❖ **Cascade Actions:** Foreign keys can specify cascade actions to determine what should happen when records in the referenced table are modified or deleted.
- ❖ Syntax: FOREIGN KEY TABLE (COLUMN) REFERENCES TABLE2(COLUMN2);
- ❖ **CASCADE DELETE:** When a referenced record is deleted, all related records in the referencing table are also automatically deleted.
- ❖ **CASCADE UPDATE:** When a referenced record's primary key is updated, the foreign key values in the referencing table are also updated accordingly.



AGGREGATE FUNCTIONS:

- SQL functions perform calculations on data.
- **Examples:** SUM (calculates the sum of values), AVG (calculates the average), COUNT (counts the number of rows).
- **Syntax:** SELECT AGG_FUN(COLUMN) FROM TABLE WHERE CONDITION;
- 1. COUNT(); Example: Select count(age) from customers;
- 2. FUN() = AVG() / SUM(): **Example:** Select fun(age) from customers where customerid=1;
- 3. FUN2() = MAX() / MIN(): **Example:** Select fun2(age) from customers;





CLAUSES:

- **GROUP BY:**

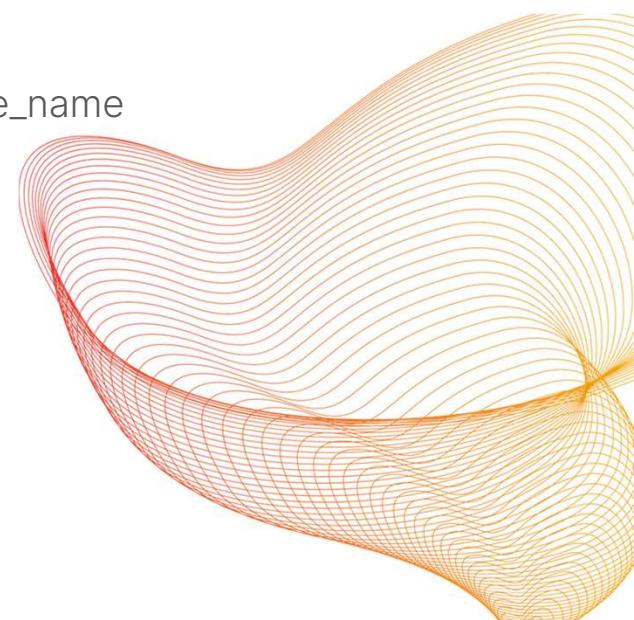
- **Syntax:** SELECT column1, aggregate_function(column2) FROM table_name
GROUP BY column1;

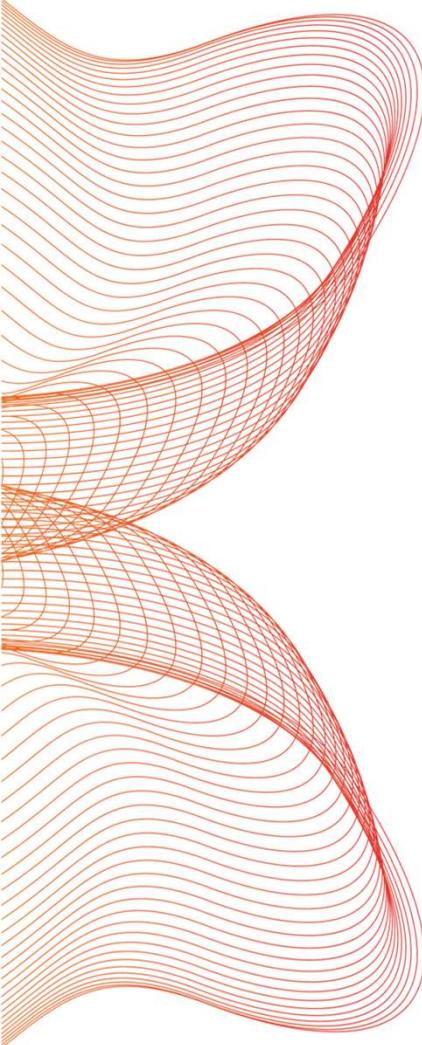
- **Example:** select sum(age) from customers group by city;

- **HAVING:**

- **Syntax:** SELECT column1, aggregate_function(column2) FROM table_name
GROUP BY column1 HAVING condition;

- **Example:** select sum(age) from customers group by city having
sum(age)>25;



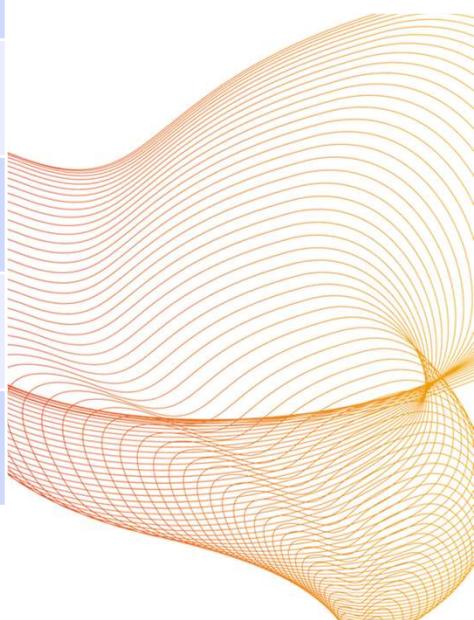


□ Consider Another Table Orders

- **Syntax:**
 - CREATE TABLE Orders (OrderID INT PRIMARY KEY, CustomerID INT, Product VARCHAR(50), Quantity INT, OrderDate DATE);
 - INSERT INTO Orders (OrderID, CustomerID, Product, Quantity, OrderDate) VALUES (1, 1, 'Laptop', 1, '2023-06-15'), (2, 1, 'Desktop', 1, '2023-07-02'), (3, 2, 'Tablet', 3, '2023-06-28'), (4, 3, 'Laptop', 1, '2023-07-10'), (5, 4, 'Smartphone', 2, '2023-06-18');
- 

 *ORDERS TABLE:*

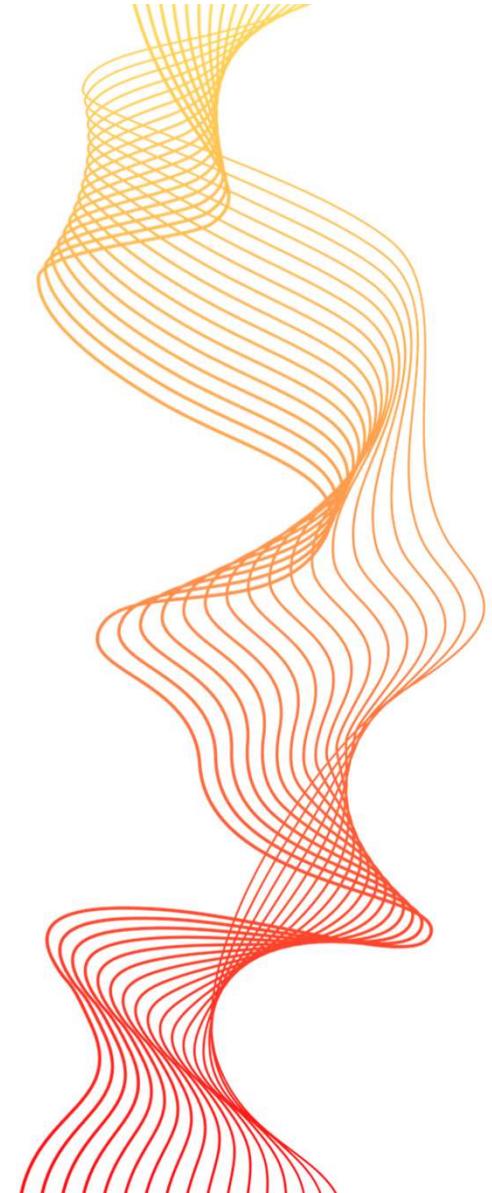
OrderID	CustomerID	Product	Quantity	OrderDate
1	1	Laptop	1	2023-06-15
2	1	Smartphone	1	2023-07-02
3	2	Desktop	3	2023-06-28
4	3	Laptop	1	2023-07-10
5	4	Smartphone	2	2023-06-18





INTRODUCTION TO JOINS:

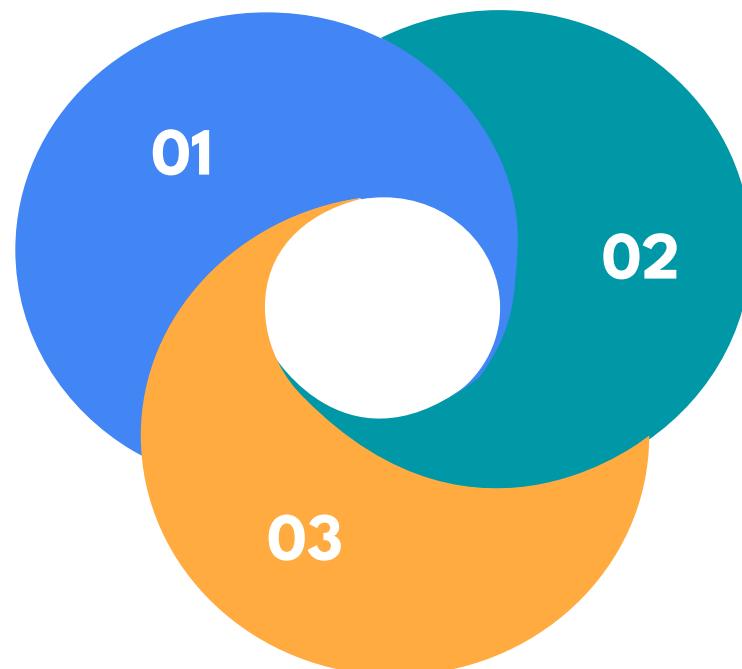
- This presentation provides an overview of SQL joins and constraints, including left join, right join, full outer join, and various types of integrity constraints.
- Joins are used to create relationships and connections between data, enabling more complex and comprehensive queries.



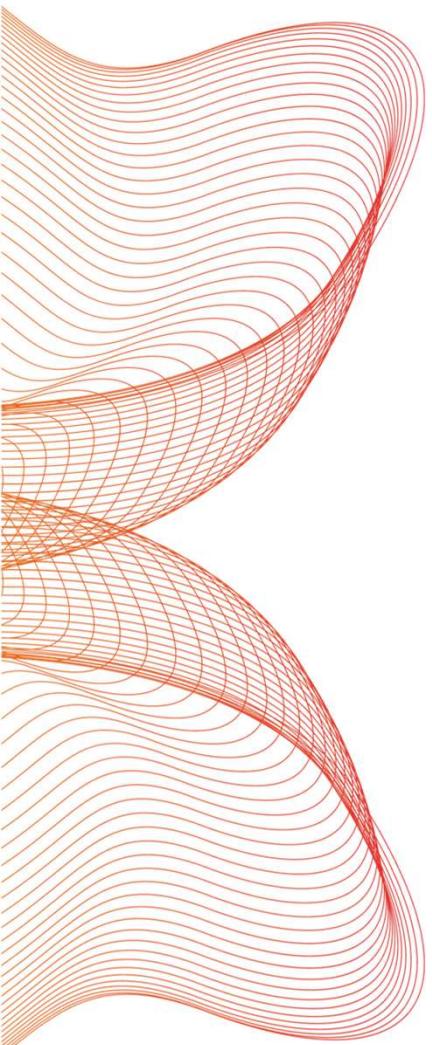
JOINS

INNER JOIN:

- ```
select c.firstname
from customers as c
inner join orders as o
on
c.customerid=o.custo
merid;
```
- \*CUSTOMERID IS THE  
FOREIGN KEY IN  
ORDERS.



- SQL joins combine rows from multiple tables based on related columns.



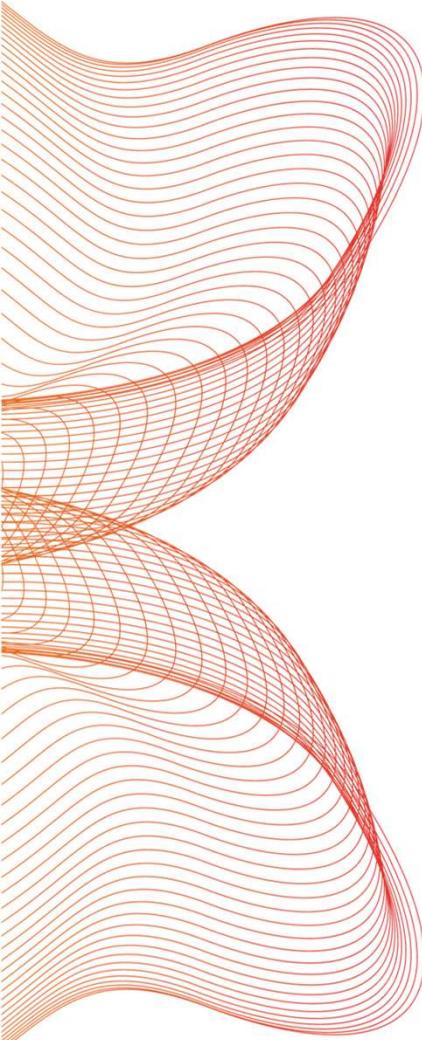
## LEFT JOIN:

**01** Left join returns all rows from the left table (customers) and the matching rows from the right table (orders).

**02** Syntax:

```
select c.city from customers as c left join orders as o
using c.customerid;
```





## **RIGHT JOIN:**

- Right join returns all rows from the right table (orders) and the matching rows from the left table (customers)
  - Syntax:  
select c.age from customers as c right  
join orders as o on  
c.customerid=o.customerid;
- 

## FULL OUTER JOIN:

- 01** Full outer join returns all rows from both tables (customers and orders), including unmatched rows.

- 02** [Syntax:](#)

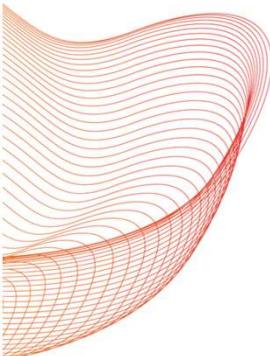
```
select c.city from customers as c full outer join orders
as o using c.customerid;
```

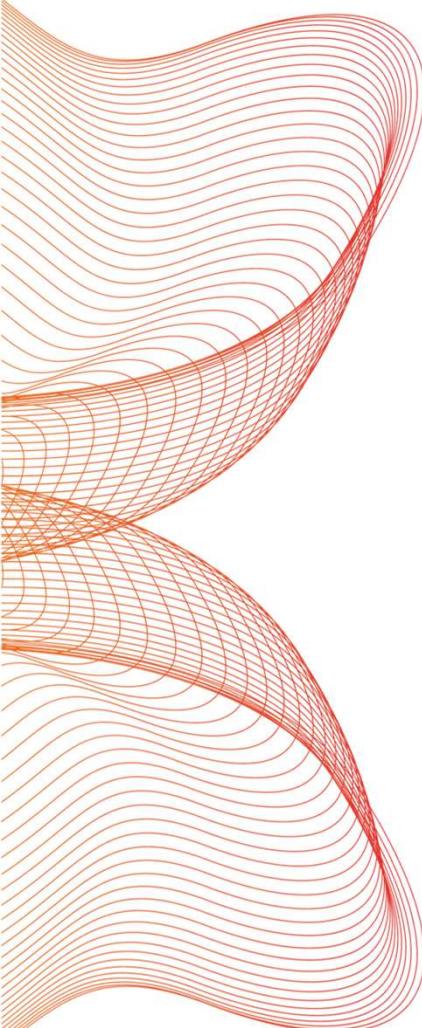




## SQL CONSTRAINTS:

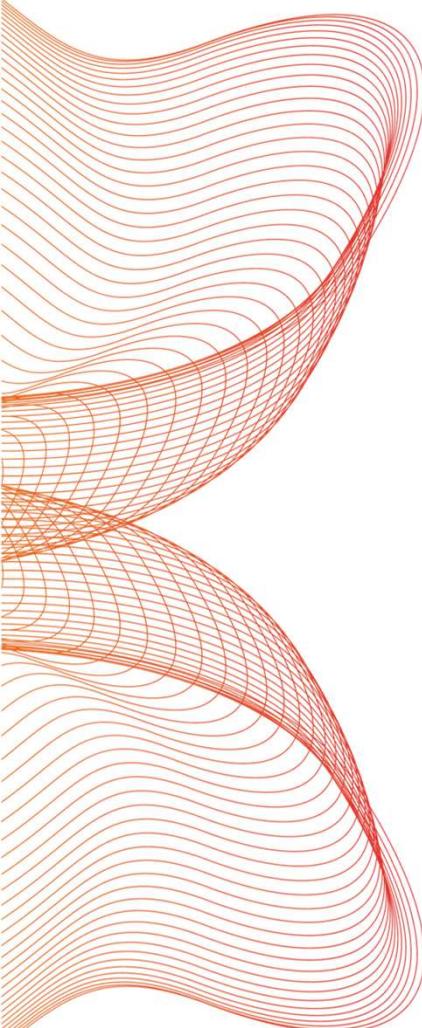
- ❖ Constraints are rules applied to tables to enforce data integrity.
- ❖ Common constraints include NOT NULL and UNIQUE.
- ❖ Types of integrity constraints are Entity Integrity Constraint, Referential Integrity Constraint, and Domain Integrity Constraint.



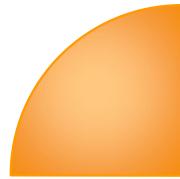


## ENTITY INTEGRITY CONSTRAINTS:

- 01** Typically enforced by defining a primary key for the table.
  
  - 02** Ensures that each row in a table is uniquely identifiable.
- 



## REFERENTIAL INTEGRITY CONSTRAINTS:

- Ensures that relationships between tables are maintained.
  - Typically enforced by using foreign keys.
  - Two types of referential integrity constraint are:
    - 1.Primary key constraint.
    - 2.Foreign key constraint.
- 

## DOMAIN INTEGRITY CONSTRAINT:

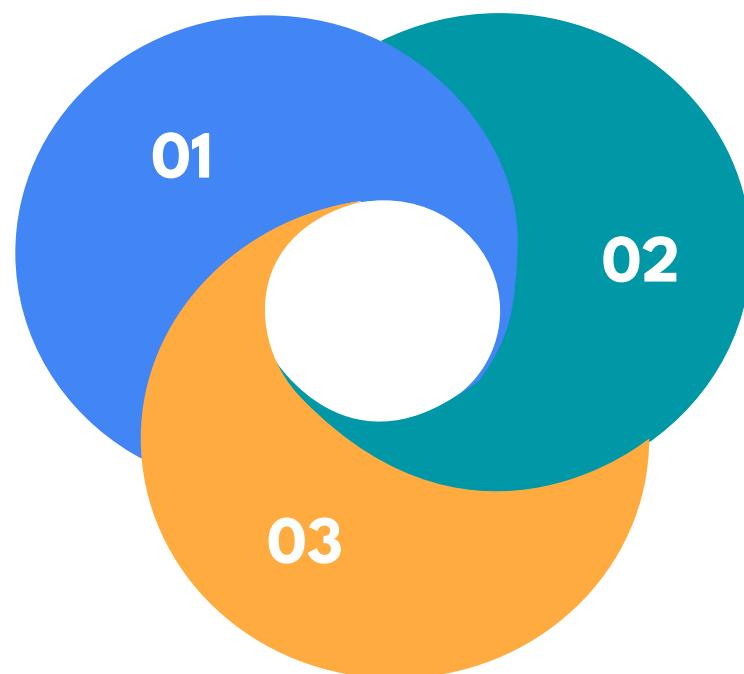
- 01** Utilizes data type definitions and constraints like CHECK.
  
- 02** Enforces the validity of data types and the range of values that can be stored in a column.



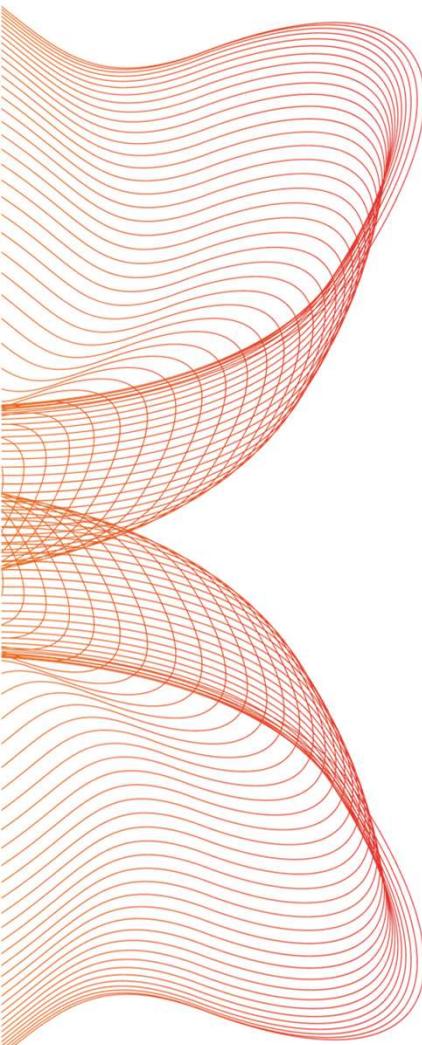
## ASSERTIONS

- ❖ Assertions in SQL allow you to define custom business rules or conditions that must be satisfied for data to be considered valid.

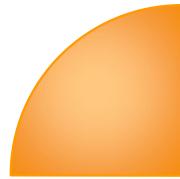
- ❖ **Syntax:** CREATE ASSERTION assertion\_name CHECK (condition)

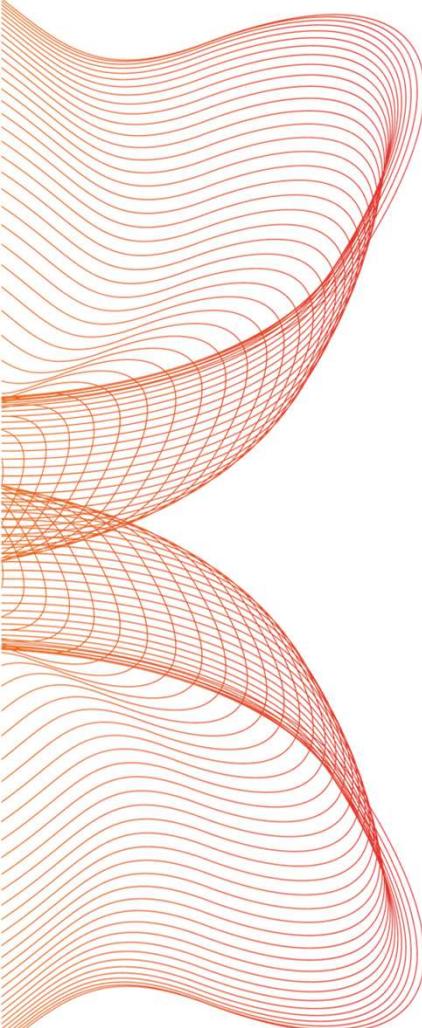


- ❖ Unlike other constraints, assertions apply to the entire database.



## *Example Assertion:*

- 01**    CREATE ASSERTION check\_order\_item\_quantity CHECK  
(NOT EXISTS (SELECT 1 FROM CUSTOMERS AS o JOIN  
ORDERS AS oi ON o.order\_id = oi.order\_id WHERE  
oi.quantity > 2))
  
  - 02**    ❖ It is to check whether condition is correct or  
incorrect.
- 



# TRIGGERS

**01**

In SQL, a trigger is a database object that automatically executes a set of actions or a stored procedure when a specific event occurs in the database.

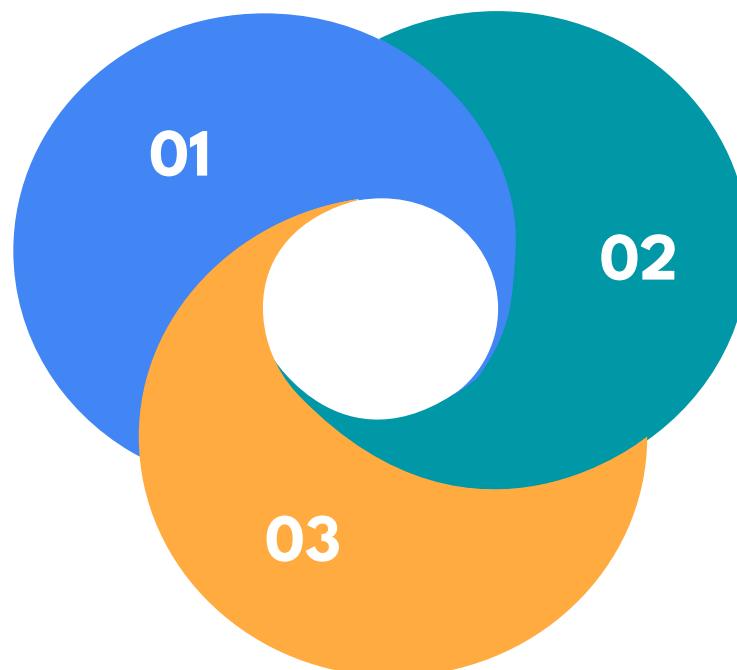
**02**

They are typically associated with specific tables and are executed in response to data manipulation operations like INSERT, UPDATE, or DELETE.



## ROW LEVEL TRIGGERS:

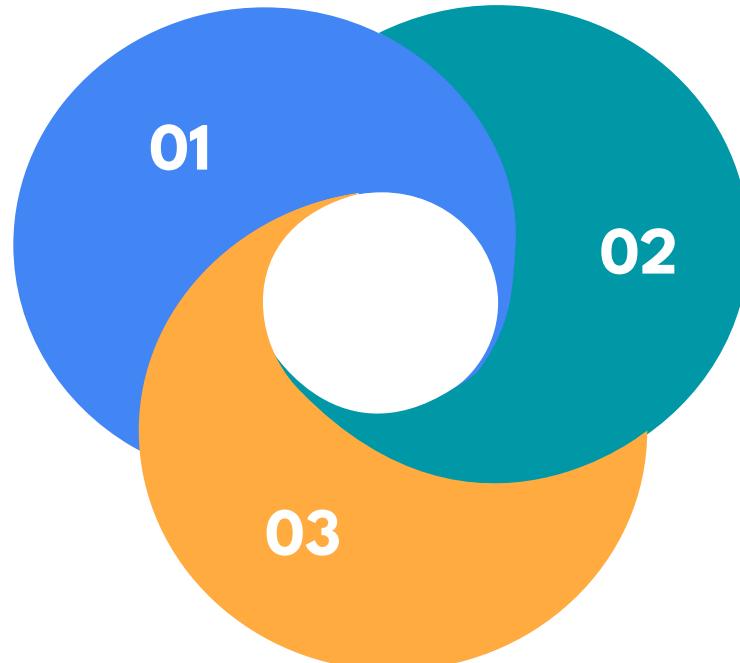
- ❖ Used for data manipulation language and not by data defined language.
- ❖ Values of attributes before and after an update can be referenced :
  - referencing old row as : for deletes and updates.
  - ✓ referencing new row as : for inserts and update.



- ❖ **SYNTAX:**

```
CREATE [OR REPLACE]
TRIGGER trigger_name
{BEFORE | AFTER}
{INSERT |
UPDATE | DELETE} ON
table_name
[FOR EACH ROW]
[WHEN (condition)]
BEGIN
END;
```

## STATEMENT LEVEL TRIGGERS

- ❖ Use for each statement instead of for each row
  - ❖ Transition tables cannot be used with before triggers, but can be used with after triggers, regardless of whether they are statement triggers or row triggers.
  - ❖ Use referencing old table or referencing new table to refer to temporary tables (called transition tables) containing the affected rows
- 



# TALENT SEARCH (JOB PORTAL)

ITW -2 PROJECT



TEAM MATES: RAKESH BANOTH  
(22075018 , SOPHOMORE FROM C.S.E, IIT BHU)

VADITHYAVATH SRISAILAM  
(22075091, SOPHOMORE FROM C.S.E, IIT BHU)

# FEATURES OF WEBSITE(TALENT SEARCH)

1. HOMEPAGE
2. USER LOGIN
3. COMPANY LOGIN
4. APPLYING FOR A JOB
5. POSTING A JOB
6. ACCEPTING/REJECTING AN APPLICANT



# FEATURES FOR USER

1. User can check the available jobs
2. They can apply for the job, they are interested in.
3. They can check whenever a new job is posted.
4. They can check the detailed description of the available jobs.
5. They will get a notification from the applied company about their application status.

# FEATURES FOR COMPANY

- 1.They can add the jobs.
- 2.Companies have privilege to edit their jobs.
- 3.They can check the applicants whoever applied to their company along with their resume.
- 4.They have full control to select or reject a particular applicant.
- 5.Moreover they can add the start and end date for applying to a certain job in their company.

# FEATURES FOR ADMIN

1. Admin have all the rights to delete an applicant.
2. Admins should accept the company before they can post the jobs in the website.
3. Admin have all the privileges to reject a company if the verification has gone wrong.
4. Admin can edit the status of the company at any time.

# HOME PAGE

Talent Search

HOME

USER LOGIN

ADMIN LOGIN

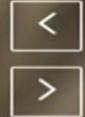
COMPANY LOGIN



Find The Best Startup Job  
That Fit You

Search A Job

Find A Talent



# SIGNUP PAGE

## Talent Search

[HOME](#)[USER LOGIN](#)[ADMIN LOGIN](#)[COMPANY LOGIN](#)

First Name

Enter First Name

Last Name

Enter Last Name

Username

Enter Username

Password

Enter Password

Confirm Password

Confirm Password

Email Id

Enter Email Id

Contact Number

Enter Contact Number

Gender

 Male     Female

Logo of the Company

Choose File | No file chosen

Company Name

Enter Name of the Company

**Submit**

# USER HOME PAGE

## Talent Search

[HOME](#)   [JOB LIST](#)   [LOGOUT](#)

Welcome Abhiram Reddy

First Name

Abhiram

Last Name

Reddy

Username

abhiram@gmail.com

Email Id

abhiram@gmail.com

Contact Number

5643217821

Gender

Male    Female

Profile Photo

[Choose File](#)   No file chosen



**Submit**

# COMPANY HOME PAGE

## Talent Search

[HOME](#)[ADD JOBS](#)[JOB LIST](#)[ALL APPLICANTS](#)[LOGOUT](#)

Welcome Unu reddy

First Name

Last Name

Username

Company Name

Email Id

Contact Number

Gender

 Male     Female

Logo of the Company

 No file chosen**Submit**

# USER JOB LIST

## Talent Search

HOME **JOB LIST** LOGOUT

Welcome Abhiram Reddy

| Sr.No | Company Name | Job Title       | Salary   | Location  | Created On    | Apply          |
|-------|--------------|-----------------|----------|-----------|---------------|----------------|
| 1     | UNR          | Data Analyst    | 100000.0 | Banglore  | Nov. 15, 2022 | <b>Apply</b>   |
| 2     | UNR          | Stack developer | 10000.0  | Hyderabad | Nov. 15, 2022 | <b>Applied</b> |

© Talent Search, All Right Reserved.

Home | Cookies | Help | FQAs

# PAGE FOR ADDING JOBS BY COMPANY

**Talent Search**

HOME    **ADD JOBS**    JOB LIST    ALL APPLICANTS    LOGOUT

Welcome Unu reddy

Job Title

Start Date

End Date

Experience (in years)

Salary (per month)

Skills Required

Company Location

Job Description

**Submit**

© Talent Search, All Right Reserved.

Home | Cookies | Help | FQAs

# JOB LIST

## Jobs Offered by a Company

### Talent Search

[HOME](#)   [ADD JOBS](#)   [JOB LIST](#)   [ALL APPLICANTS](#)   [LOGOUT](#)

Welcome Unu reddy

| Sr.No | Job Title       | Created On    | Action | Delete |
|-------|-----------------|---------------|--------|--------|
| 1     | Stack developer | Nov. 15, 2022 |        |        |
| 2     | Data Analyst    | Nov. 15, 2022 |        |        |

# APPLICANTS APPLIED FOR A COMPANY

## Talent Search

[HOME](#) [ADD JOBS](#) [JOB LIST](#) [ALL APPLICANTS](#) [LOGOUT](#)

Welcome Unu reddy

| Sr.No | Job Title       | Applicant | Applied On    | Resume | Delete |
|-------|-----------------|-----------|---------------|--------|--------|
| 1     | Stack developer | Abhiram   | Nov. 15, 2022 |        |        |

© Talent Search, All Right Reserved.

[Home](#) | [Cookies](#) | [Help](#) | [FAQs](#)

# ADMIN INTERFACE FOR ALL COMPANIES

## Talent Search

COMPANY ▾    VIEW APPLICANTS    LOGOUT    Welcome

| Sr.No | Full Name      | Email Id             | Contact    | Gender | Company Name | Image                                                                                | Status   | Change Status                  | Delete                  |
|-------|----------------|----------------------|------------|--------|--------------|--------------------------------------------------------------------------------------|----------|--------------------------------|-------------------------|
| 1     | Unu reddy      | unureddy@gmail.com   | 8371813792 | Male   | UNR          |   | Rejected | <button>Change Status</button> | <button>Delete</button> |
| 2     | Nikhil Neerudu | nnlwgl2003@gmail.com | 7416724547 | Male   | NIK          |   | Accepted | <button>Change Status</button> | <button>Delete</button> |
| 3     | Unu Reddy      | unureddy@gmail.com   | 8371813792 | Male   | UU           |  | Accepted | <button>Change Status</button> | <button>Delete</button> |

© Talent Search. All Right Reserved.

Home | Cookies | Help | FQAs

# ADMIN INTERFACE FOR ALL APPLICANTS

## Talent Search

[COMPANY ▾](#)[VIEW APPLICANTS](#)[LOGOUT](#)

Welcome

| Sr.No | Full Name      | Email Id          | Contact    | Gender | Image                                                                                 | Action                 |
|-------|----------------|-------------------|------------|--------|---------------------------------------------------------------------------------------|------------------------|
| 1     | Nikhil Neerudu |                   | 7416724547 | Male   |    | <a href="#">Delete</a> |
| 2     | kiran kumar    | kiran@gmail.com   | 1234567890 | Male   |   | <a href="#">Delete</a> |
| 3     | Abhiram Reddy  | abhiram@gmail.com | 5643217821 | Male   |  | <a href="#">Delete</a> |

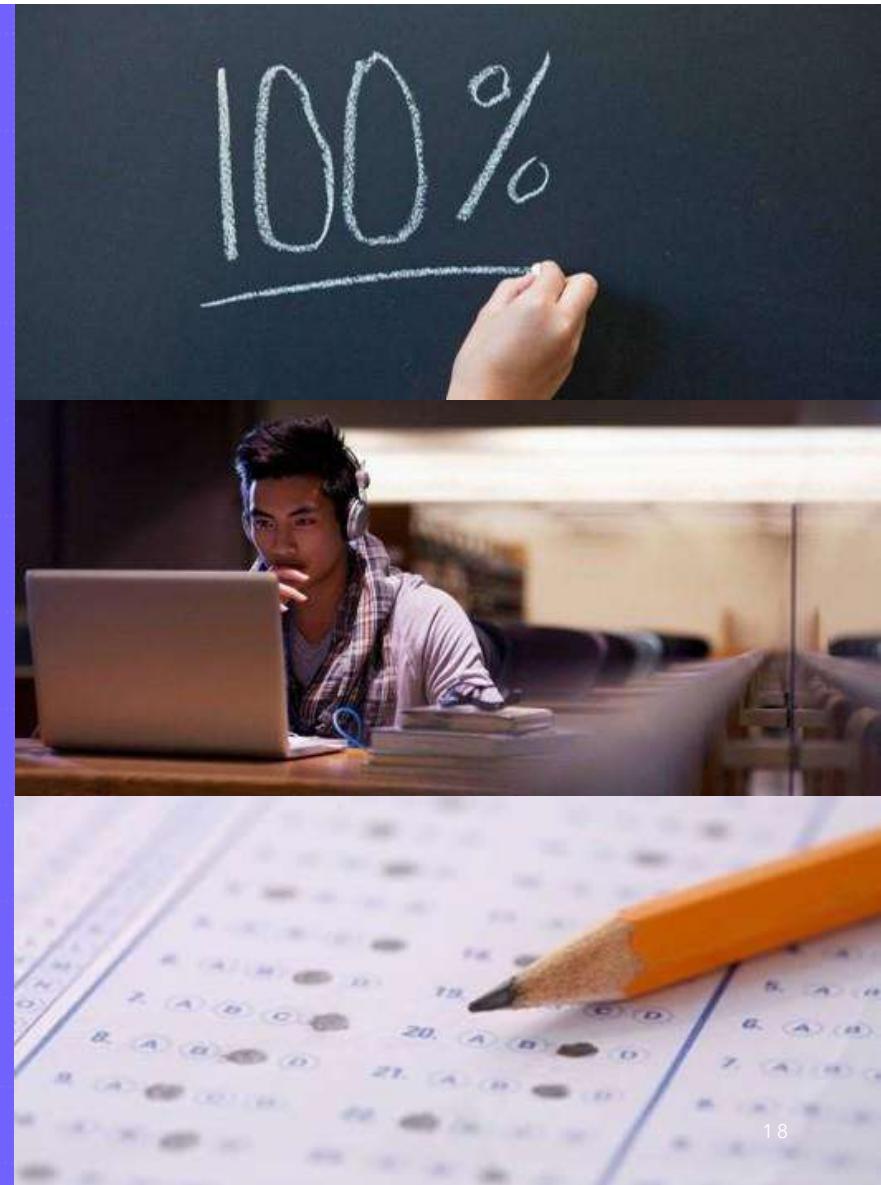


WE PROVIDE YOU A BETTER  
INTERFACE FOR FINDING THE BEST  
JOB THAT FITS YOU.

- TALENT SEARCH.

# Summary

Basically, we have prepared a web application that can help users from various industries to look for a job which suits them the best , in the similar way we have added an interface to the companies who are recruiting the staff in their esteemed companies. So, this will help in two ways for both the users who are looking for a job and for the companies who are recruiting the staff members.



# Thank you

-BANOTH RAKESH (22075018)

-VADITHYAVATH SRISAILAM (22075091)

