## **Basic Data Definition Language (DDL) - Oracle**

Some commonly used commands for Data Definition Language (DDL) in Oracle are:

**CREATE TABLE:**

-- Syntax:

CREATE TABLE table\_name (

column\_name1 data\_type1,

column\_name2 data\_type2,

...

);

-- Example:

CREATE TABLE employees (

employee\_id NUMBER(10) PRIMARY KEY,

name VARCHAR2(50) NOT NULL,

salary NUMBER(10,2)

);

This command creates a new table named "employees" with three columns: "employee\_id" (a unique identifier), "name", and "salary".

**ALTER TABLE:**

-- Syntax:

ALTER TABLE table\_name

ADD column\_name data\_type;

-- Example:

ALTER TABLE employees

ADD department\_id NUMBER(10);

This command adds a new column named "department\_id" to the "employees" table.

**DROP TABLE:**

-- Syntax:

DROP TABLE table\_name;

-- Example:

DROP TABLE temporary\_table;

This command deletes the table named "temporary\_table" from the database.

**CREATE INDEX:**

-- Syntax:

CREATE INDEX index\_name ON table\_name (column\_name);

-- Example:

CREATE INDEX emp\_name\_idx ON employees (name);

This command creates an index named "emp\_name\_idx" on the "name" column of the "employees" table.

**DROP INDEX:**

-- Syntax:

DROP INDEX index\_name;

-- Example:

DROP INDEX emp\_name\_idx;

This command deletes the index named "emp\_name\_idx" from the database.

**CREATE VIEW:**

-- Syntax:

CREATE VIEW view\_name AS

SELECT column\_list

FROM table\_name

WHERE condition;

-- Example:

CREATE VIEW sales\_report AS

SELECT product\_name, SUM(sales) AS total\_sales

FROM sales

GROUP BY product\_name;

This command creates a view named "sales\_report" that shows the total sales for each product.

**DROP VIEW:**

-- Syntax:

DROP VIEW view\_name;

-- Example:

DROP VIEW sales\_report;

This command deletes the view named "sales\_report" from the database.

## **Basic Data Manipulation Language (DML) - Oracle**

Some examples of commonly used Data Manipulation Language (DML) commands in Oracle, along with their syntax and a brief explanation:

**INSERT:**

-- Syntax:

INSERT INTO table\_name (column\_list)

VALUES (value\_list);

-- Example:

INSERT INTO employees (employee\_id, name, salary)

VALUES (100, 'John Doe', 50000);

This command inserts a new row into the "employees" table with the specified values for the "employee\_id", "name", and "salary" columns.

**UPDATE:**

-- Syntax:

UPDATE table\_name

SET column\_name = value

WHERE condition;

-- Example:

UPDATE employees

SET salary = salary \* 1.10

WHERE department\_id = 10;

This command updates the salary of all employees in the department with ID 10 by increasing it by 10%.

**DELETE:**

-- Syntax:

DELETE FROM table\_name

WHERE condition;

-- Example:

DELETE FROM employees

WHERE employee\_id = 100;

This command deletes the employee with ID 100 from the "employees" table.

**SELECT:**

-- Syntax:

SELECT column\_list

FROM table\_name

WHERE condition;

-- Example:

SELECT employee\_id, name, salary

FROM employees

WHERE department\_id = 10;

This command retrieves the employee ID, name, and salary for all employees in the department with ID 10.

**ORDER BY:**

-- Syntax:

SELECT column\_list

FROM table\_name

ORDER BY column\_name;

-- Example:

SELECT employee\_id, name, salary

FROM employees

ORDER BY salary DESC;

This command retrieves the employee ID, name, and salary for all employees, sorted in descending order by salary.

**WHERE:**

-- Syntax:

SELECT column\_list

FROM table\_name

WHERE condition;

-- Example:

SELECT employee\_id, name, salary

FROM employees

WHERE salary > 50000;

This command retrieves the employee ID, name, and salary for all employees with a salary greater than 50000.

## **JOIN Operation**

Certainly! Here are some examples of commonly used JOIN operations in Oracle, along with their syntax and a brief explanation:

**INNER JOIN:**

-- Syntax:

SELECT column\_list

FROM table1

INNER JOIN table2

ON table1.column\_name = table2.column\_name;

-- Example:

SELECT e.employee\_id, e.name, d.department\_name

FROM employees e

INNER JOIN departments d

ON e.department\_id = d.department\_id;

This command performs an INNER JOIN between the "employees" (e) and "departments" (d) tables, linking employees to their respective departments based on the "department\_id" column. It retrieves the employee ID, name, and department name for all employees.

**LEFT JOIN:**

-- Syntax:

SELECT column\_list

FROM table1

LEFT JOIN table2

ON table1.column\_name = table2.column\_name;

-- Example:

SELECT e.employee\_id, e.name, d.department\_name

FROM employees e

LEFT JOIN departments d

ON e.department\_id = d.department\_id;

This command performs a LEFT JOIN between the "employees" (e) and "departments" (d) tables. It retrieves the employee ID, name, and department name for all employees, even if they do not have a corresponding entry in the "departments" table. In such cases, the department name will be NULL.

**RIGHT JOIN:**

-- Syntax:

SELECT column\_list

FROM table1

RIGHT JOIN table2

ON table1.column\_name = table2.column\_name;

-- Example:

SELECT e.employee\_id, e.name, d.department\_name

FROM employees e

RIGHT JOIN departments d

ON e.department\_id = d.department\_id;

This command performs a RIGHT JOIN between the "employees" (e) and "departments" (d) tables. It retrieves the employee ID, name, and department name for all employees in the "departments" table, even if they do not have a corresponding entry in the "employees" table. In such cases, the employee ID and name will be NULL.

**FULL OUTER JOIN:**

-- Syntax:

SELECT column\_list

FROM table1

FULL OUTER JOIN table2

ON table1.column\_name = table2.column\_name;

-- Example:

SELECT e.employee\_id, e.name, d.department\_name

FROM employees e

FULL OUTER JOIN departments d

ON e.department\_id = d.department\_id;

This command performs a FULL OUTER JOIN between the "employees" (e) and "departments" (d) tables. It retrieves all rows from both tables, even if they do not have a corresponding entry in the other table. In such cases, the missing values will be NULL.

These are just a few examples of the various JOIN operations available in Oracle. For more information and a complete list of JOIN types, refer to the Oracle documentation.