DDL

- **Creating a Table**:

```
CREATE TABLE table_name_2 (
        Col_name_1  VARCHAR(20) NOT NULL,
        Col_name_2 DATE,
        Col_name_3 NUMBER NOT NULL,
        PRIMARY KEY (Col_name_1, Col_name_3),
        FOREIGN KEY (Col_name_1) REFERENCES table_name_1
);
```

Note: Copy fk from ONE to MANY side

- **Adding and dropping an index to speed up retrieval**

```
CREATE INDEX idx_name ON table_name (col1, col2);
DROP INDEX index_name ON table_name;
```

- **Drop Schema components**

DROP SCHEMA schema_name CASCADE

Drop the entire schema including all tables.

CASCADE option deletes all data, all tables, indexes, domains, etc.

DROP SCHEMA schema_name RESTRICT

Removes the schema only if it is empty.

DROP TABLE table_name

Remove the table and all of its data.

DROP TABLE table_name CASCADE

Remove the table and all related tables as specified by FOREIGN KEY constraints.

DROP TABLE table_name RESTRICT

Remove the table only if it is not referenced (via a FOREIGN KEY constraint) by other tables.

- **Alter Table (used to modify columns)**

```
ALTER TABLE table_name_2
        ADD CONSTRAINT fk_table_name
        FOREIGN KEY (col_name) REFERENCES table_name_1 (col_name);
ALTER TABLE table_name
        DROP PRIMARY KEY;
ALTER TABLE table_name
        ADD COLUMN col_name datatype;
ALTER TABLE table_name
        ADD COLUMN column_name datatype;
ALTER TABLE table_name
        DROP COLUMN column_name;
ALTER TABLE table_name
        ALTER COLUMN column_name datatype;
```

DML

- **Inserting Data into Tables**

```
INSERT INTO table_name (column1, column2, ... columnX)
    VALUES (val1, val2, ... valX);
```

- **Select**

```
SELECT   column1, column2, ... columnN
FROM     tableA, tableB, ... tableZ
WHERE    condition1, condition2, ...conditionM
GROUP BY column1, ...
HAVING   condition
ORDER BY column1, column2, ... column
```

- **Inner Join**

```
SELECT  employee.first_name, employee.last_name, department.department_name
FROM    employee INNER JOIN department
    ON  employee.departmentid = department.departmentid
WHERE department.department_location = 'NY';
```

- **Outer Join**

A **RIGHT JOIN** (or RIGHT OUTER JOIN) will favor the table listed on the right hand side so that all of its records will be shown.

A **LEFT JOIN** (or LEFT OUTER JOIN) will favor the table listed on the left hand side so that all of its records will be shown.

- **Update Table (used to modify rows)**

  the UPDATE command is used to change attribute values in the database.

  ```
  UPDATE table_name
   SET col_name = '___'
   WHERE col_name(key) = 'xx';


  UPDATE employee
  SET salary = salary * 1.05
  WHERE employeeid = 'E101';
  ```

- **Delete Tuples**

  DELETE is used to remove tuples from a table.

  With no WHERE clause, DELETE will remove all tuples from a table.

  ```
     DELETE FROM employee
     WHERE salary > 50000;


     DELETE FROM employee
     WHERE departmentid IN
       (SELECT departmentid
        FROM   department
        WHERE  department_location = 'NY');
  ```

- **SQL Views**

  ```
  CREATE VIEW view_name AS
  Your_query


  CREATE VIEW avg_sal_dept AS
  SELECT department, AVG(salary)
  ```

```
FROM   employee

GROUP BY department;
```