Data Definition Language (DDL) Statements

The DROP TABLE statement removes an existing table with all its data from the database and moves it to the recycle bin
After dropping a table, we can restore it for a short time using the FLASHBACK TABLE statement.
After dropping a table, all the objects related to that table will also be deleted or become invalid.

```
DROP TABLE employees_copy4;

FLASHBACK TABLE employees_copy4 TO BEFORE DROP;
```

DROP+TABLE+Statement(Code+Samples).sql

Data Definition Language (DDL) Statements

EraaSoft

- TRUNCATE TABLE Statement& The DELETE statement deletes all data row by row whereas the TRUNCATE statement deletes all rowfrom a table more quickly

- The TRUNCATE statement is one of the DDL (DataDefinition Language) statements so it willauto-commit changes immediately after removing data.

- TRUNCATE does not allow rollback.

- The data deleted using the TRUNCATE statement cannot easily be restored (FLASHBACK) becauseTRUNCATE does not generate any undo information or log data.

- The TRUNCATE statement works faster than the DELETE statement.

By Eslam Khder

Data Definition Language (DDL) Statements

```sql
SELECT * FROM employees_copy;
DELETE FROM employees_copy;
TRUNCATE TABLE employees_copy;
DROP TABLE employees_copy;

CREATE TABLE employees_test AS SELECT * FROM employees;

SELECT COUNT(*) FROM employees_test;

DELETE FROM employees_test;

TRUNCATE TABLE employees_test;

DROP TABLE employees_test;
```

TRUNCATE+TABLE+Statement+(Code+Samples).sql

By Eslam Khder

EraaSoft

RENAME Statement

The RENAME statement is used to change the name of an existing column or table

We can change the name of a column.

We can change the name of a table.

Data Definition Language (DDL) Statements

```
DESC employees_copy;
ALTER TABLE employees_copy RENAME COLUMN hire_date TO start_date;

RENAME employees_copy TO employees_backup;

SELECT * FROM employees_copy;
SELECT * FROM employees_backup;

ALTER TABLE employees_backup RENAME TO employees_copy;
SELECT * FROM employees_copy;
```

RENAME+Statement+(Code+Samples).sql

Data Definition Language (DML) Statements

DML is used to add, update, and delete data.

A collection of DML statements is called a transaction.

A transaction starts with the first execution of a DML statement and finishes with a commit or rollback.

By Eslam Khder

Data Definition Language (DML) Statements

INSERT STATEMENT

Use to insert row in table.

By Eslam Khder

Data Definition Language (DML) Statements

```
INSERT INTO jobs_copy (job_id, job_title, min_salary, max_salary)
VALUES('PR_MGR', 'Project Manager', 7000, 18000);

INSERT INTO jobs_copy (job_title, min_salary, job_id, max_salary)
VALUES('Architect',6500,'ARCH',15000);

INSERT INTO jobs_copy
VALUES('DATA_ENG','Data Engineer',8000,21000);

INSERT INTO jobs_copy (job_id, job_title, min_salary)
VALUES('DATA_ARCH','Data Architecture',8000);

ALTER TABLE jobs_copy MODIFY max_salary DEFAULT 10000;

INFO jobs;

INSERT INTO jobs_copy (job_id, job_title, min_salary)
VALUES('DATA_ARCH2','Data Architecture2',8000);

INSERT INTO jobs_copy (job_id, min_salary)
VALUES('DATA_ARCH2',8000);
```

INSERT+Statement+(Part+1)+(Code+Samples).sql

By Eslam Khder

Data Definition Language (DML) Statements

```sql
INSERT INTO jobs_copy
VALUES('DATA_ARCH2','Data Architecture2',8000);

INSERT INTO jobs_copy
VALUES('DATA_ARCH3','Data Architecture3',8000, NULL);

SELECT * FROM employees_copy;

INSERT INTO employees_copy SELECT * FROM employees;

INSERT INTO employees_copy SELECT * FROM employees WHERE job_id = 'IT_PROG';

INSERT INTO employees_copy(first_name,last_name,email,hire_date,job_id)
SELECT first_name,last_name,email,hire_date,job_id FROM employees WHERE job_id = 'IT_PROG';
```

INSERT+Statement+(Part+2)+(Code+Samples).sql

By Eslam Khder

Data Definition Language (DML) Statements

UPDATE STATEMENT

Use to UPDATE row in table.

By Eslam Khder

Data Definition Language (DML) Statements

```sql
DROP TABLE employees_copy;
CREATE TABLE employees_copy AS SELECT * FROM employees;

SELECT * FROM employees_copy;

UPDATE employees_copy
SET salary = 500;

SELECT * FROM employees_copy WHERE job_id = 'IT_PROG';

UPDATE employees_copy
SET salary = 50000
WHERE job_id = 'IT_PROG';

UPDATE employees_copy
SET salary = 5, department_id = null
WHERE job_id = 'IT_PROG';

UPDATE employees_copy
SET (salary, commission_pct) = (SELECT max(salary), max(commission_pct) FROM employees)
WHERE job_id = 'IT_PROG';

UPDATE employees_copy
SET     salary     = 100000
WHERE   hire_date = (SELECT MAX(hire_date) FROM employees);
```

UPDATE+Statement+(Code+Samples).sql

By Eslam Khder

Data Definition Language (DML) Statements

DELETE STATEMENT

Use to DELETE row from table.

By Eslam Khder

Data Definition Language (DML) Statements

```sql
SELECT * FROM employees_copy;

DELETE FROM employees_copy;

DELETE employees_copy;

DELETE employees_copy
WHERE job_id = 'IT_PROG';
```

DELETE+Statement+(Code+Samples).sql

By Eslam Khder

Using+SELECT+Statements

```sql
SELECT * FROM employees;
```

Using+SELECT+Statements(Code+Samples).sql

```sql
SELECT first_name, last_name, email FROM EMPLOYEES;
```

```sql
SELECT * FROM employees;
SELECT * FROM departments;
```

SQL+Statement+Basics(Code+Samples).sql

By Eslam Khder

Using Column Aliases

```sql
SELECT first_name, last_name, email FROM employees;
SELECT first_name AS name, last_name as surname, email FROM employees;
SELECT first_name AS "My     Name", email "E-mail" FROM employees;
SELECT first_name AS "My Name", email "E-mail" FROM employees;
SELECT employee_id, salary + nvl(salary*commission_pct,0) + 1000 new_salary, salary FROM employees;
```

Using+Column+Aliases(Code+Samples).sql

By Eslam Khder

DISTINCT and UNIQUE Operators

```sql
SELECT first_name FROM employees;
SELECT distinct first_name FROM employees;
SELECT unique first_name FROM employees;
SELECT distinct job_id, distinct department_id FROM employees;
SELECT distinct job_id, department_id FROM employees;
SELECT distinct job_id FROM employees;
SELECT distinct job_id, department_id, first_name FROM employees;
SELECT job_id, distinct department_id, first_name FROM employees;
```

DISTINCT+and+UNIQUE+Operators(Code+Samples).sql

By Eslam Khder

Concatenation Operators

```sql
SELECT 'My Name is Alex' FROM employees;
SELECT 'My Name is ' || first_name FROM employees;
SELECT 'The commission percentage is ' || commission_pct AS concatenation,commission_pct FROM employees;
SELECT first_name || ' ' || last_name AS "full name" FROM employees;
SELECT * FROM employees;
SELECT * FROM locations;
SELECT street_address || ',' || city || ',' || postal_code || ',' || state_province || ',' || country_id AS "full address"
FROM locations;
```

Concatenation+Operators(Code+Samples).sql

By Eslam Khder

EraaSoft

Arithmetic Opertors

```sql
SELECT * FROM employees;
SELECT employee_id, salary, salary*12 as annual_salary FROM employees;
SELECT employee_id, salary, salary+100*12 as annual_salary FROM employees;
SELECT employee_id, salary, (salary+100)*12 as annual_salary FROM employees;
SELECT sysdate FROM dual;
SELECT sysdate + 4 FROM dual;
SELECT employee_id, hire_date, hire_date+5 FROM employees;
SELECT salary, salary*commission_pct, commission_pct FROM employees;
```

Arithmetic+Opertors+and+NULL+values(Code+Samples).sql

By Eslam Khder

EraaSoft

Using WHERE Clause

```sql
SELECT * FROM employees;
SELECT * FROM employees WHERE salary > 10000;
SELECT * FROM employees WHERE job_id = 'IT_PROG';
```

Using+WHERE+Clause(Code+Samples).sql

By Eslam Khder