

## SQL Chapter 5 Statements FSN

**Statements** - to create, fire table & insert data.

↳ 3 types of statements

DDL	DML	TCL	DCL
Data Definition Language [CDTAR]	Data Manipulation Language [IUD]	Transaction Control [RCST]	Data Control Language [GR]
Create Drop Truncate Alter Rename	Insert Update Delete	Rollback Commit Savepoint	Grant Revoke
- Auto Commit ✗ No Rollback	Needed Commit ✓ ⑥ Roll back option		
deals with Structure	deals with Data		on permission to database

## Data Types

1) Char  
→ stores fixed length character data  
- alphanumeric ✓  
Ex: Name char(6)

2) Varchar/2  
→ stores variable length character data  
- alphanumeric  
Ex: Name varchar(6)

T	O	M	-	-	-
---	---	---	---	---	---

Blank space/  
Reserved / Non Reusable  
memory

max 2k charach

T	O	M	.	.	.	.
---	---	---	---	---	---	---

Null  
Reusable memory

max 4k charach

### 3) Number

- stores numeric data

Ex: Sal number(4). → 9999

\_\_\_\_\_ (6,2)

9999.99  
precision  
scale

\_\_\_\_\_ (4,3)

9.999  
.

— (2,2)

.99

### 4) DATE

- stores date & time  
→ need to specify length

ex: order\_date date,

default format:

dd-month-yy

### 5) BLOB

Binary Large Object

- stores binary data.  
(images, movies,  
audiofiles) in database

### 6) CLOB

Character Large Object

- stores plain character  
data like verche

upto  
4GB

## DDL ①

### → CREATE

(Ex):

CREATE TABLE products

prodid	NUMBER(4)
prodname	VARCHAR(10)
qty	NUMBER(3)
description	VARCHAR(20)

products (

prodid	NUMBER(4) PRIMARY KEY,
prodname	NOT NULL,
qty	CHECK (qty > 0)
description	VARCHAR(20)

);

... will comment fields in other table

Kontenurung einer Tabelle mit einer Referenz

Erl:

Ex: products (prodid)  
prodid NUMBER(4) REFERENCES products (prodid)

Creating table from another table

```
CREATE TABLE temp  
AS  
SELECT * FROM employees;
```

Note: temp copies only ALL COLUMNS, ~~ALL ROWS~~ ✓  
& NOT NULL constraints ✓

NEVER COPIES ~~PK, FK, check constraints~~ X

Qn?  
I have a table with 1M records,  
How do I duplicate it into another table w/o  
using insert & w/o inserting  
individually all records to table?

(DDL)

2) TRUNCATE

Removes all the data permanently, but structure of the table remains as it is

Ex: TRUNCATE TABLE temp;

⇒ op no rows will display  
records removed  
structure/desc ✓

(DDL 3)

3) DROP

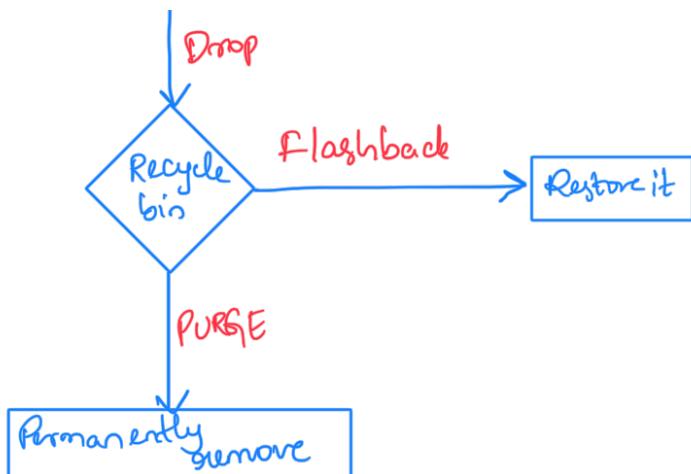
removes both data & structure of table permanently  
from the database

Ex: DROP TABLE temp;

⇒ op table doesn't exist

106 Recycle Bin

Table



## RENAME

RENAME TABLE1 TO TABLE3;

## ALTER

↳ query alter/ changes structure of the table

- add column/s
- remove column/s
- rename column/s
- ... etc

### ① add new column

Alter table **product**

add **model-no**

**varchar(10) NOT NULL**;

### ② drop existing column

Alter table **products**

drop **model-no** ;

### 3) rename column

Alter table **products**

    Rename column **qty** to **qty-avail**;

Select ~~not DML~~, ~~DML~~, not belong to any group bcz  
 it does not define }  
 not manipulate }  
 it just display date  
 what is required by user

## **DML**

**Insert** - inserts a record to a table;

Insert into **table-name**

values **( 1001, 'string' )**

} values to be inserted  
as per constraints  
data types

add on  
sysdate

if child table f.f.  
record must be in  
parent table

other way to add specific column:

insert into **table-name (column1, column5)**

values **( 03, 'Sal' )**;

## UPDATE

↳ updates one or more records:

Ex:

```
UPDATE [emp_table] SET sal=300, comm=400  
      WHERE emp_id=102;  
      This is optional.
```

## DML DELETE

- delete one/some/all records

Ex:

```
delete from [employees]  
      WHERE emp_id=121;
```

## TCL

### Transaction Control language

- Any DML change is not permanent
- We need to save the DML changes in order to make it permanent
- we can also undo/ignore some DML changes on a table

TCL

## ROLLBACK

- it undoes the DML changes performed on a table

- **Rollback**

- rollbacks before DML change

## COMMIT

- It saves the DML changes permanently to the database

- Note!

Committing after rollback } → will not have  
Rollback after commit } any effect

Note 2:

During abnormal exit = shutdown / SQL window close. by mouse click.

→ All DMLs will be **rolled back** automatically

During

Normal exit =

- All DMLs will be **auto committed** &  
- there will be no rollback

Eg:

Commit / Rollback

Insert

- No Auto Commit

Update

- No Auto Commit

Alter

- Auto Commit ↓ saved till now

Delete

- No Auto Commit

Rollback

-   
      ↙  
      roll back before  
      del

Insert

↑

Update

} No Auto Commit

Delete

All a rolled back

Rollback

**SAVEPOINT**

- It's like a pointer (or breakpoint)  
till where DML can be rolled back

Ex: `Save point X;` ←  
update  
delete  
`Rollback to X;`

### Composite Primary Key

- primary key created using more than 1 column

Ex: `alter table library`  
add primary key (`reg_no, book_no, DOI`)