SQL CORE Query   
  
1) View All Data 🡪 SELECT \* FROM "MainDataBase";

2) Limit the Number of rows 🡪 SELECT \* FROM "MainDataBase" LIMIT 5 or 10;

3) To find any specific row record 🡪SELECT \* FROM "MainDataBase" WHERE "APN" = '400-Clutch';

4) Search by partial data🡪SELECT \* FROM "MainDataBase" WHERE "Description" LIKE '% ----%';

5) To find All Parts Owned by a Specific person🡪SELECT \* FROM "MainDataBase" WHERE "Owner" = ' nikhil16';

6) To get count of data in the table🡪 SELECT COUNT(\*) FROM "MainDataBase";

7) To find data which values are missing🡪 SELECT \* FROM "MainDataBase" WHERE "Email" IS NULL;(suppose email is null)

8) To get selected records like APN, Name, Owner 🡪SELECT "Name", "Type", "Owner" FROM "MainDataBase";

9) To keep the data in order like new data first🡪SELECT \* FROM "MainDataBase"

ORDER BY "CreatedDate" DESC;

10) To filter the multiple values (use IN) 🡪 SELECT \* FROM "MainDataBase" WHERE "Owner" IN ('coder22', 'nikhil16');

11) To filter the data using rang🡪 SELECT \* FROM "MainDataBase"

WHERE "CreatedDate" BETWEEN '2025-08-09 00:00' AND '2025-08-11 00:00';

12) to filter with multiple conditions🡪SELECT \* FROM "MainDataBase"

WHERE "Type" = 'Transmission' AND "Owner" = 'coder22';

13) Sort the data in alphabetical order🡪SELECT \* FROM "MainDataBase"

ORDER BY "Name" ASC;

14) Search by objected🡪SELECT \* FROM "MainDataBase" WHERE "ObjectId" = 3;

15) To delete a record based on where condition🡪 DELETE FROM table\_name

WHERE condition;

16) Drop… Delete an entire table in the db.🡪 DROP TABLE table\_name;

17) To delete all row data without a where clause uses truncate🡪 TRUNCATE TABLE table\_name;

18) To update a data in the table 🡪 UPDATE table\_name SET column1 = value1, column2 = value2, ... WHERE condition;

19) To create a backup table before drop🡪 CREATE TABLE backup\_maindatabase AS TABLE "MainDataBase";

20) Updating selected multiple columns 🡪 UPDATE "MainDataBase"

SET "APN" = CASE "ObjectId"

WHEN 13 THEN '400-Clutch'

WHEN 9 THEN '702-GearBox'

ELSE "APN"

END

WHERE "ObjectId" IN (13, 9);

22-08-2025

1. Trigger creation

How to create a trigger function 🡪 CREATE OR REPLACE FUNCTION entrydate()RETURNS TRIGGER AS $$

BEGIN INSERT INTO audit("ObjectId", "EntryDate") VALUES (NEW.ObjectId, NOW());

RETURN NEW;END;

$$ LANGUAGE plpgsql;

1. Trigger creation to main data table 🡪 CREATE TRIGGER trg\_entrydate

AFTER INSERT ON audit

FOR EACH ROW

EXECUTE FUNCTION entrydate();

1. Insert values into the table 🡪 INSERT INTO "MainDataBase"("APN", "Name", "SuperType", "Type", "Description", "CreatedDate", "Owner", "Email")

VALUES ('802-Caliper', '802-001-APN', 'Part', 'Braking System', 'Car calipers', '2025-08-11 10:52', 'nikhil16', 'nikhil16@apn.com');

1. To create an fts\_document 🡪 alter table backup\_maindatabase add column Fts\_document tsvector generated always as (to\_tsvector('english',coalesce("Name",'') ||' '|| coalesce("Type",'')||''|| coalesce("Description",''))) stored;
2. Creating an index of gin 🡪 ate index idx\_fts\_backup\_maindatabase on backup\_maindatabase using gin(fts\_document);
3. Full text search for using query 🡪 select \* from backup\_maindatabase where fts\_document @@plainto\_tsquery('english','701');
4. To create a view of the table for particular column🡪 create view backup\_maindatabase\_search as select "ObjectId","Name","Type","Description",fts\_document from backup\_maindatabase;
5. Basic search with multiple keywords🡪 select \* from table\_name where fts\_document @@ plainto\_tsquery(‘english’, ‘system 701’);
6. Search with OR (|) 🡪 SELECT \* FROM "MainDataBase" WHERE fts\_document @@ to\_tsquery('english', 'braking | system');
7. Negation (exclude a word)🡪 select \* from table\_name where fts\_document @@ to\_tsquery(‘english’, ‘braking & ! Calipers’);
8. Ranking result 🡪SELECT \*, ts\_rank(fts\_document, plainto\_tsquery('english', 'braking system')) AS rank FROM "MainDataBase" WHERE fts\_document @@ plainto\_tsquery('english', 'braking system') ORDER BY rank DESC;
9. Column renaming🡪 ALTER TABLE table\_name RENAME COLUMN current\_column\_name TO new\_column\_name;
10. Adding a column in the table🡪 ALTER TABLE table\_name ADD COLUMN new\_column\_name column\_type;
11. Returning data on insert Using the ***RETURNING*** clause, you can immediately retrieve data from the newly inserted record🡪 Insert into table\_name(“Name”,”Type”,”Description”) values(‘aa’,’bb’,’abcds’) Returning id;
12. Deleting bulk data🡪 delete from table\_name where id= any(‘{1,2,3,4}’);
13. Getting random rows🡪select \* from table\_name order by random() limit 5 or 10;
14. Sorting of multiple columns asc or desc🡪select column1, column2 from table\_name order by column\_name asc/desc;