

Experiment No: 06

Title: To create nested queries and view for the given Database.

Batch: A2 Roll No.: 16010421063 Experiment No: 06

Aim: To create nested queries and view for the given database (Virtual lab).

Resources needed: PostgreSQL PgAdmin3

Theory:

Nested subqueries:

in clause:

The in connective tests for the set membership, where the set is a collection of values produced by a select clause.

For example to select details of the books written by r.p.jain and d.perry use

select book_id, book_name,price from book where author in(,,r.p.jain", ,,d. perry","godse");

not in:

This connective tests for absence of the set membership.

For example to select details of the books written by authors other than r.p.jain and d.perry use

select book_id, book_name,price from book where author not in(,,r.p.jain", ,,d. perry","godse");

all:

this keyword is basically used in set comparison query.

It is used in association with relational operators.

"> all" corresponds to the phrase "greater than all".

For example to display details of the book that have price greater than all the books published in year 2000 use.

Select book_id, book_name, price from book where price >all (select price from book where pub_year="2000");

any or some:

These keywords are used with relational operators in where clause of set comparison query.

exists and not exists:

exists is the test for non empty set. It is represented by an expression of the form 'exists (select From) '. Such expression evaluates to true only if the result evaluating the subquery represented by the (select From) is non empty. for example to select names of the books for which order is placed use

select book_name from book where exists(select * from order where book_id=order.book_id);

[&]quot;=some" is identical to in and ">some" is identical to not in.

[&]quot;>any " is nothing but "greater than at least one".

(Autonomous College Affiliated to University of Mumbai)

Views:

Views are virtual tables created from already existing tables by selecting certain columns or certain rows. A view can be created from one or many tables. View allows to,

- Restrict access to the data such that a user can only see limited data instead of complete table.
- Summarize data from various tables which can be used to generate reports.

In PostgrSQL, Views are created using the CREATE VIEW statement given bellow.

CREATE [TEMP | TEMPORARY] VIEW view_name AS SELECT column1, column2..... FROM table name WHERE [condition];

For example,

Consider COMPANY table having following records:

id | name | age | address | salary

----+------+-----+------

- 1 | Paul | 32 | California | 20000
- 2 | Allen | 25 | Texas | 15000
- 3 | Teddy | 23 | Norway | 20000
- 4 | Mark | 25 | Rich-Mond | 65000
- 5 | David | 27 | Texas | 85000
- 6 | Kim | 22 | South-Hall | 45000
- 7 | James | 24 | Houston | 10000

Following statement creates a view from COMPANY table.

CREATE VIEW COMPANY_VIEW AS SELECT ID, NAME, AGE FROM COMPANY; Now, query can be written on COMPANY VIEW in similar way as that of an actual table, as

shown below,

SELECT * FROM COMPANY VIEW;

This would produce the following result:

View can be dropped using "DROP VIEW" statement.

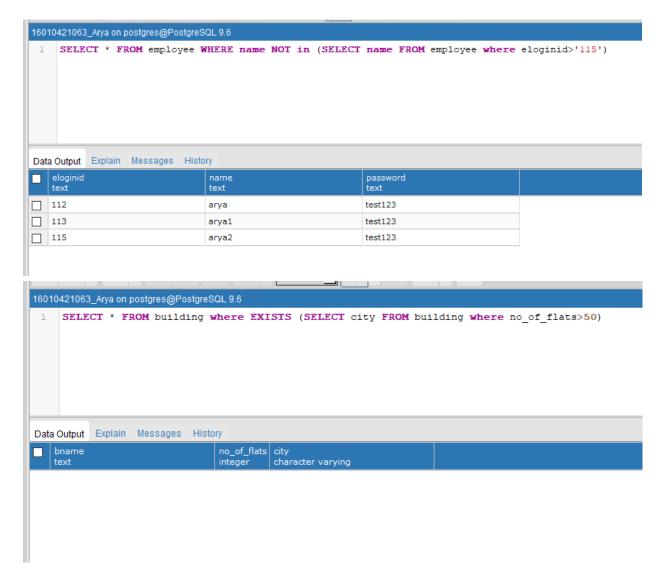
Procedure / Approach / Algorithm / Activity Diagram:

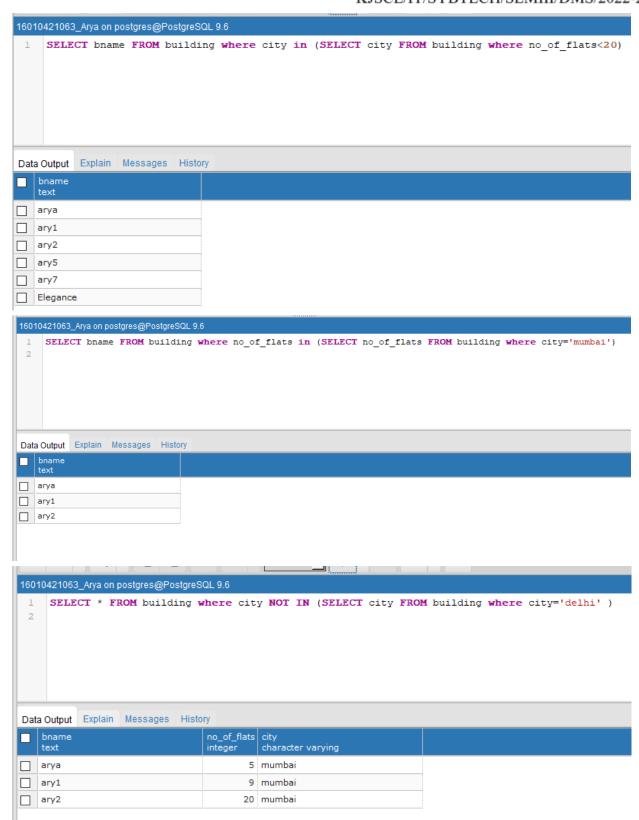
- 1 Refer different syntax given in theory section and formulate queries consisting of nested sub queries, in , not in, as, group by, having etc clauses and different set operations for your database.
- 2 Create views from existing tables Execute SELECT, UPDATE, INSERT statements on views and original table.

Results: (Program printout with output / Document printout as per the format)

1

Nested Subqueries





VIEWS

16010421063_Arya on postgres@PostgreSQL 9.6 2 CREATE VIEW building_name_no_view AS SELECT bname,no_of_flats FROM building_name_no_view AS SELECT bname_no_view AS SELECT 3 SELECT * FROM building name no view Explain Messages History Data Output bname no_of_flats text integer arya 5 9 ary1 20 ary2 ary5 50 ary7 50 Elegance 8 16010421063_Arya on postgres@PostgreSQL 9.6 1 2 CREATE VIEW building_name_city_view AS SELECT bname,city FROM building; SELECT * FROM building_name_city_view Explain Messages History Data Output character varying arya mumbai ary1 mumbai ary2 mumbai delhi ary5 delhi ary7 Elegance delhi

| 160 | 10421063_Arya on postgres@Postgres | SQL 9.6 | |
|-----|--|---|--------------------|
| 1 2 | | _view as select eloginid ,n a ame_view | ame FROM employee; |
| Dat | a Output Explain Messages Historical Explain Explain Messages Historical Explain Messa | name | |
| | text | text | |
| | | | |
| | | arya | |
| | 113 | arya1 | |
| | | • | |

Outcomes:

CO3. Illustrate the concept of security, Query processing, indexing and Normalization for Relational database.

Questions:

1. Explain what are the disadvantages using view on update function.

Ans. Disadvantages:

- 1. When a table is dropped, associated view become irrelevant.
- 2. Since view are created when a query requesting data from view is triggered, its bit slow
- 3. When views are created for large tables, it occupy more memory.
- 2 Can we use where clause with group by clause? Justify your answer A WHERE clause is used to filter rows based on a specified condition. When we use GROUP BY clause in conjugation with WHERE clause, the WHERE clause filters the rows first based on the mentioned condition and then GROUP BY clause prepares a summary row only for the filtered rows. It becomes very useful when we want to GROUP only specific rows together.
- 3 Can we use having and group by clause without Aggregate functions? Justify your answer
 - That's how GROUP BY works. It takes several rows and turns them into one row. Because of this, it has to know what to do with all the combined rows where there have different values for some columns (fields). This is why you have two options for every field you want to SELECT: Either include it in the GROUP BY clause, or use it in an aggregate function so the system knows how you want to combine the field.



KJSCE/IT/SYBTECH/SEMIII/DMS/2022-23

Conclusion: (Conclusion to be based on the objectives and outcomes achieved) Understood the concept of nested queries and views and implemented the same.

Grade: AA / AB / BB / BC / CC / CD /DD Signature of faculty in-charge with date

References:

Books/ Journals/ Websites:

- 1. Korth, Slberchatz, Sudarshan, :"Database System Concepts", 6th Edition, McGraw Hill
- 2. Elmasri and Navathe, "Fundamentals of Database Systems", 5thEdition, PEARSON Education

