Section 1: Simple - But Powerful - SQL Statements

1. Join Our Community!

2. Course Resources

3. What is PostgreSQL All About?

* PostgreSQL is a powerful, open source object-relational database system.
* PostgreSQL is used as the primary data store or data warehouse for many web, mobile, geospatial, and analytics applications.
* Challenges of postgres
  + Writing efficient queries to retrieve

Information.

* Designing the schema, or structure,

of the database.

* Understanding when to use

advanced features

* Managing the database in a

production environment

4. Database Design

* Database Design Process
  + What kind of thing are we storing?
  + What properties does this thing have?
  + What type of data does each of those properties contain?

5. Creating tables

* Online SQL editor : <https://pg-sql.com/>
* To create a table : CREATE TABLE <table-name> (<column-name> <DATATYPE>, …);

6. Analyzing CREATE TABLE

* Keywords : Always written in Capital letters. (Ex. CREATE, TABLE,…)
* Identifiers : Always written in lower case letters. (Ex. Cities,name,…)

7. Inserting Data Into a Table

* To insert data into table : INSERT INTO <table-name> (<column-name>,…) VALUES (<values>)
* To insert multiple data into table : INSERT INTO <table-name> (<column-name>,…) VALUES (<values>),(<values>),…,(<values>);

8. Retrieving Data with Select

* To retrieve data from table : SELECT column1, column2, … FROM <table-name>;
* To retrieve all data from table : SELECT \* FROM <table-name>;

9. Calculated Columns

* We can use mathematical operators in select statement to obtain calculates columns in result.
* To display proper name in result use 'AS' statement.

10. Calculating Phone Revenue

11. Exercise Solution

12. String Operators and Functions

* String operators and functions

|  |  |
| --- | --- |
| || | Join two strings |
| CONCAT ( ) | Join two strings |
| LOWER ( ) | Gives a lower case string |
| LENGTH ( ) | Gives number of characters in a string |
| UPPER ( ) | Gives an upper case string |

* We can use string constants using quotation symbol.

Section 2: Filtering Records

13. Filtering Rows with "Where"

* Syntax :

SELECT column1, column2, ...

FROM table\_name

WHERE condition;

14. More on the "Where" Keyword

* Where keyword options

|  |  |
| --- | --- |
| = | Are the values equal? |
| > | Is the value on the left greater? |
| < | Is the value on the left less? |
| >= | Is the value on the left greater or  equal to? |
| IN | Is the value present in a list? |
| <= | Is the value on the left lesser or equal to? |
| <> | Are the values not equal ? |
| != | Are the values not equal ? |
| BETWEEN | Is the value between two other values ? |
| NOT IN | Is the value not present in a list ? |

15. Compound "Where" Clauses

* Use case of the where keyword options.

16. A "Where" Exercise Overview

17. A "Where" Solution

* SELECT name, price FROM phones

WHERE units\_sold > 5000;

18. "Where" With Lists

19. A "Where" With Lists Solution

* SELECT name, manufacturer

FROM phones

WHERE

manufacturer = "Apple’ OR manufacturer = 'Samsung';

20. Calculations in "Where" Clauses

* We can use mathematical operators in where statement to obtain calculated columns in result.

21. Solving Calculations

* SELECT

name,

price \* units\_sold AS

total\_revenue

FROM

phones

WHERE

price \* units\_sold > 1000000;

22. Updating Rows

* UPDATE table\_name

SET column1 = value1, column2 = value2, ...

WHERE condition;

23. Deleting Rows

* DELETE FROM table\_name WHERE condition;

24. A Solution for Updating Rows

25. Solution for Deleting Rows

Section 3: Working with Tables

26. The Plan Moving Forward

27. Approaching Database Design

* For a complicated application, database designing is a crucial part.
* Ex: For a photo sharing application design includes (users,likes,comments,photos,etc).

28. One-to-Many and Many-to-One Relationships

* One-to-Many is the most commonly used relationship among tables. A single record from one table can be linked to zero or more rows in another table.
* Many-to-One relationship in DBMS is a relationship between more than one instances of an entity with one instance of another entity.

29. One-to-One and Many-to-Many Relationships

* One-to-One (1-1) relationship is defined as the relationship between two tables where both the tables should be associated with each other based on only one matching row.
* Many-to-Many relationship lets you relate each row in one table to many rows in another table and vice versa.

30. Primary Keys and Foreign Keys

* The PRIMARY KEY constraint uniquely identifies each record in a table.
* A FOREIGN KEY is a field (or collection of fields) in one table, that refers to the PRIMARY KEY in another table.

31. Understanding Foreign Keys

* There can be multiple foreign keys in a table relating to other tables.

32. Auto-Generated ID's

* In postgres we use SERIAL keyword after id to generate unique id's then add PRIMARY KEY keyword.

33. Creating Foreign Key Columns

* In postgres for foreign key we use : <column-name> <datatype> REFERENCES <table-name>(<column-name>)

34. Running Queries on Associated Data

* Join queries are used to combine data of two table and display a preferred result.

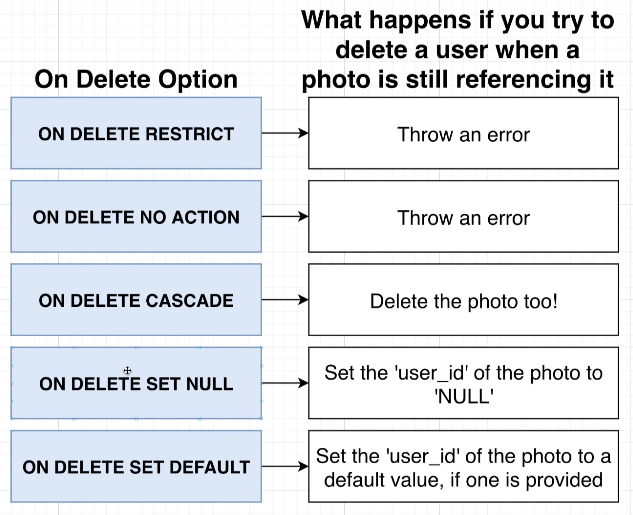
35. Exercise Overview

36. Foreign Key Creation Solution

37. Foreign Key Constraints Around Insertion

* If we put a random foreign key which doesn't corelate to anything then it will show an error.
* "NULL" is used instead of random number for foreign key which doesn't need to be related.

38. Constraints Around Deletion



39. Commands You’ll Need for the Next Video

40. Testing Deletion Constraints

* "DROP" command is used drop a table.
* Use the above delete options while creating the table.

41. Setting Foreign Keys to Null on Delete

* Use the ON DELETE SET NULL option to set the user id to NULL when user Is deleted.

42. Adding Some Complexity

* Adding complex database design for the example photo sharing application.