UNIT3: ADVANCED CONCEPTS OF DATA AND DATABASES

DATA TYPES AND TABLES

LAGUARDIA COMMUNITY COLLEGE

Fun Fact: The concept of relational databases was born out of a groundbreaking paper titled "A Relational Model of Data for Large Shared Data Banks," published in 1970 by Edgar F. Codd, an English computer scientist working at IBM. Codd was seeking a more efficient way to store and retrieve data in databases, which at the time were hierarchical and cumbersome to manage. His revolutionary idea was to use "relations," now known as tables, to represent data, which could then be linked or related based on common key values. This approach not only simplified data management but also introduced a level of abstraction that made databases more flexible and accessible. Codd's relational database model fundamentally changed the landscape of data storage and retrieval, laying the groundwork for the SQL language and the modern databases we rely on today. His work earned him the prestigious Turing Award in 1981, recognizing his profound impact on the field of computer science.

A self JOIN is a regular join, but the table is joined with itself.

Self joins are used when you want to combine rows with other rows from the same table.

To perform the self join, you must use a table alias so SQL can determine which is the LEFT and RIGHT table from the same table.

SELF JOINS

Self-Joins

```
SELECT
f1.title AS Film1,
f2.title AS Film2,
c.name AS category
FROM
Film_category fc1
JOIN
Film_category fc2 ON fc1.category_id = fc2.category_id AND
fc1.film_id < fc2.film_id
JOIN category c ON fc1.category_id = c.category_id
JOIN
film f1  ON fc1.film_id = f1.film_id
JOIN film f2 ON fc2.film_id = f2.film_id
ORDER BY
c.name, f1.title;
```

SELF JOINS

Previously, inner and outer joins were used to help deal with combining data from multiple tables.

Self join allows you to refer to the same table twice – as if it were two separate tables.

The self join essential creates a virtual view of a table, allowing it to be used more than once.

SELF-JOIN

- A subquery is the optimal query – especially for performance.
- Always use an aliases (AS)

```
SELECT
    f1.title AS "Film 1",
    f2.title AS "Film 2",
    f1.rating AS "Rating"
FROM
    film f1
INNER JOIN
    film f2 ON f1.rating = f2.rating AND f1.film_id < f2.film_id
ORDER BY
    f1.rating, f1.title;
```

OVERVIEW OF DATA TYPES

BOOLEAN

CHARACTER

NUMBER

TEMPORAL (DATE, ETC.)

SPECIAL TYPES

ARRAY

DATA TYPE - BOOLEAN

Booleans hold two values – True or False. If the data is unknown, then the NULL value is used.

In order to declare a column that consists of a Boolean value, use the Boolean or bool keyword.

Data that is inserted into a Boolean column will be convert into a Boolean value – for example – Yes, True, Y, T, 1 are converted to True. 0, no. N, No, F are converted to False

Boolean Examples

```
SELECT
customer_id,
first_name || ' ' || last_name AS customer_name
FROM
Customer
WHERE
activebool = TRUE;
```

DATA TYPE - CHARACTER

- Char – a single character
- **Character Data Types**. Stores strings of letters, numbers, and symbols. **Data types CHARACTER** (CHAR) and **CHARACTER VARYING** (VARCHAR) are collectively referred to as **character** string **types**, and the values of **character** string **types** are known as **character** strings.

DATA TYPE – CHAR VS. VARCHAR

CHAR is a conceptually fixed-length, blank-padding string. Trailing blanks (or spaces) are removed on input, but restored on output.

VARCHAR is a variable-length character data type. The default length is 80.

Normally, VARCHAR is used for all string data. CHAR is used when you need fixed-width string output.

CHAR Examples

```
SELECT title, rating
FROM film
WHERE rating = 'PG-13';
```

DATA TYPE - NUMERIC

Integers

Floating-poin
t numbers

INTEGERS

Small integer (smallint) is 2-byte, signed integer (both positive & negative) with a range of -32768 to 32767

Integer (int) is a 4-byte int with -2147483648 to 2147483647.

Serial – same as int, however similar to AUTO_INCREMENT in other databases, it populates values into columns automatically.

FLOATING-POINT NUMBERS

Float( ) – can hold 8 bytes of data, or 15 places after the decimal point.

Real or float8 is a double-precision (8-byte) floating-point number

Numeric or numeric(p,s) is a real number with p digits with s numbers after the decimal point.

Numeric Examples

```
SELECT title, release_year
M film
WHERE release_year = 2005;
```

DATA TYPE – TEMPORAL

- Temporal store date and time data
  - Date stores date data
  - Timestamp stores date and time
  - Time stores time
  - Interval stores difference in timestamps

Temporal Examples

```
SELECT rental_id, rental_date, customer_id, inventory_id
FROM rental
WHERE rental_date BETWEEN '2005-07-01' AND '2005-07-31';

SELECT rental_id, rental_date, return_date
FROM rental
WHERE rental_date BETWEEN '2005-07-01' AND '2005-07-31'
AND return_date <= '2005-08-01';
```