Single Table Queries from Mountain View Community Hospital

Ny'Chadrialynne "Chay" Malone, McKenzie Harris,& Romello Turner

Selecting Data from a Single Table

In the creation of a single-table query, essential SQL clauses are employed for selection criteria.

- SELECT
- FROM
- WHERE

Every query must include the SELECT and FROM clauses, with the WHERE clause utilized to retrieve specific rows of data. The WHERE clause provides a finer level of detail in data retrieval.

Selecting Data from a Single Table

- SELECT
  - Lists columns (attributes) from base tables, temporary tables, or custom made views.
  - Result of the SELECT query is a table.
  - Can be passed to another SQL query where a set is required.

- FROM
  - Identifies what table to pull in the columns in the SELECT statement from
  - More than one or multiple tables may be defined in this clause.

Selecting Data from a Single Table

- WHERE
  - Includes the conditional constraints used to filter what is selected.
  - Conditional statements used to introduce greater logic into relational data querying include, AND, IN, NOT, BETWEEN.

Selecting Data from a Single Table

- Simplifications
  - SELECT
- Tables
  - WHERE
- Columns
  - FROM
- Other Conditional Arguments

Select Table Example

```
SELECT PHYSICIAN_ID,PHYSICIAN_NAME FROM PHYSICIAN_2023;
```

Output for the Select Statement

PHYSICIAN_ID	PHYSICIAN_NAME
1234	Kamala Harris
1235	Antony Blinken
1236	Janet Yellen
1237	Lloyd Austin
1238	Merrick Garland
1239	Deb Haaland

Where Statements

To return the patients with a bed number less than or equal to 0004, which would input:

```
SELECT PATIENT_NO,

PATIENT_NAME, BED_NO FROM PATIENT_2023

WHERE BED_NO<=0004;
```

Where Statements

- The output from the WHERE example is as follows:

PATIENT_NO	PATIENT_NAME	BED_NO
1010	Joe Biden	0001
1011	Tom Vilsack	0002
1012	Gina Raimondo	0003
1013	Julie Su	0004

Compound Conditions: OR

Compound Conditions is when two or more conditional statements are expressed in the WHERE clause, and are separated by a conditional operator such as AND, OR, or NOT, for example.

When OR is used either condition state will be true. Such as:

```
SELECT PATIENT_NO, PATIENT_NAME, PHYSICIAN_ID, BED_NO FROM
PATIENT_2023

WHERE PHYSCIAN_ID = 1234

OR BED_NO<=0003;
```

Compound Conditions: OR

- The output from the OR example is as follows:

PATIENT_NO	PATIENT_NAME	PHYSICIAN_ID	BED-NO
1010	Joe Biden	1234	1
1011	Tom Vilsack	1234	2
1012	Gina Raimondo	1239	3

Compound Conditions: And

If the AN operator is used, both of the conditions must be true for a row to be selected.

An example of the AND operator is:

```
SELECT PATIENT_NO,PATIENT_NAME,PHYSICIAN_ID,
BED_NO FROM PATIENT_2023 WHERE PHYSICIAN_ID=1234
AND BED_NO <=0003;
```

Compound Conditions: And - Output

- The output from the Compound Condition: AND example is as follows:

PATIENT_NO	PATIENT_NAME	PHYSICIAN_ID	BED_NO
1010	Joe Biden	1234	1
1011	Tom Vilsack	1234	2

Compound Condition: Not

When the NOT operator is used, its Boolean logic is followed by the opposite of the condition to be met.

An example of this would be as follows:

```
SELECT PATIENT_NO,PATIENT_NAME,PHYSICIAN_ID

BED_NO FROM PATIENT_2023 WHERE NOT (PHYSICIAN_ID=1234);
```

Compound Condition: Not

- The output from the example is as follows:

PATIENT_NO	PATIENT_NAME	PHYSICIAN_ID	BED_NO
1009	Crystal Waters	1230	3
1012	Gina Raimondo	1239	3
1013	Julie Su	1235	4
1014	Xavier Becerra	1235	5
1015	Marcia Fudge	1236	6

Query Sorting Results

The ORDER BY clause followed by the WHERE clause will show the order of results displayed by a query.

Example:

```
SELECT * FROM EMPLOYEE_2023 WHERE EMP_NO>2000

ORDER BY EMP_NAME;
```

DESC can be used after the ORDER BY clause to sort into descending order. The default is ascending order.

Query Sorting Results

-Example output

EMP_NO	EMP_NAME
8765	Carlisa Sample
6543	Cody McGrew
5549	Josh McKinley
3852	Robert Wood
4492	Steve McNabb

Query Sorting Results: Filtering

There can be multiple levels of sorting that can be done by the primary sort key and secondary sort key. These keys are differentiated by sequential order in the ORDER BY clause.

Example:

```
SELECT * FROM PATIENT_2023 WHERE PHYSICIAN_ID>1234
ORDER BY BED_NO,PATIENT_NAME;
```

Query Sorting Results: Filtering

- Example Output.
- If BED_NO had the same value, PATIENT_NAME would then be in ascending order.

PATIENT_NO	PATIENT_NAME	PHYSICIAN_ID	BED_NO
1012	Gina Raimondo	1239	3
1013	Julie Su	1235	4
1014	Xavier Becerra	1235	5
1015	Marcia Fudge	1236	6

Aggregate Functions

SQL standards contain special functions to calculate aggregate values on data sets which are then applied groups in rows that will meet a common criteria.

Examples:

- COUNT - counts number of occurrences
- AVG - calculates the average value of a column
- SUM - calculates the total number of values in a column group
- MAX - returns the maximum value in a column group
- MIN - returns the minimum value in a column group

Depending on the Relational Database Management System use of these functions may vary.

Ranges for Selection Qualification

- Comparison operators are used to establish a range of values.
- <> (!= in some RDBMS) meaning Boolean NOT
- BETWEEN (alternative to using two comparison operators)
    - Lets user specify a range of values in a condition
- NOT BETWEEN (opposite of BETWEEN)
- Example:
    - SELECT *FROM PATIENT_2023 WHERE BED_NO<>0003;

Output for Comparison Operator<>

- BED_NO = 0003 aren't included in the following output.

PATIENT_NO	PATIENT_NAME	PHYSICIAN_ID	BED_NO
1010	Joe Biden	1234	1
1011	Tom Vilsack	1234	2
1013	Julie Su	1235	4
1014	Xavier Becerra	1235	5
1015	Marcia Fudge	1236	6

Project Exercise 4A:

- Information from only one of the tables
- SELECT EMP_NO, EMP_NAME FROM EMPLOYEE_2023 ORDER BY EMP_NO;

Output 4A:

EMP_NO	EMP_NAME
1812	Brent Wheaton
1987	Greg Shick
3852	Robert Wood
4492	Steve McNabb
5549	Josh McKinley
6534	Cody McGrew
8765	Carlissa Sample

Exercise 4B:

- Aggregate information from one attribute in the table. (scalpel)

```
SELECT COUNT (*)
    FROM ITEM_2023
        WHERE DESCRIPTION='scalpel';
```

Output 4B:

- There are 0 number of rows on the ITEM_2023 table where DESCRIPTION =’Scalpel’

COUNT(*)
0

Project Exercise 4C:

Various functions: Querying minimum unit cost from ITEM_2023

```
SELECT MIN (UNIT_COST) FROM ITEM_2023;
```

Output 4C:

MIN (UNIT COST)
1.50

Simple Query (Input):

```
SELECT PHYSICIAN_ID, PATIENT_NAME

FROM PATIENT_2023;
```

Simple Query(Output):

PHYSICIAN_ID	PATIENT_NAME
1234	Joe Biden
1234	Tom Vilsack
1239	Gina Raimondo
1235	Julie Su
1235	Xavier Becerra
1236	Marcia Fudge

Simple Query with Ordering (Input):

```
SELECT PHYSICIAN_ID, PATIENT_NAME

FROM PATIENT_2023

ORDER by PATIENT_NAME;
```

Simple Query with Ordering (Output):

PHYSICIAN_ID	PATIENT_NAME
1239	Gina Raimondo
1235	Joe Biden
1235	Julie Su
1234	Marcia Fudge
1234	Tom Vilsack
1236	Xavier Becerra

Query with > (Input):

```
SELECT WARD_NO, ROOM_NO

FROM BED_2023

WHERE ROOM_NO>200

ORDER BY ROOM_NO;
```

Query with > (Output):

WARD_NO	ROOM_NO
102	205
101	304
103	402
106	501

Simple Query(Input):

```
SELECT PATIENT_NO, ITEM_NO, QUANTITY

FROM CONSUMES_2023

ORDER BY PATIENT_NO;
```

Simple Query(Output):

PATIENT_NO	ITEM_NO	QUANTITY
1010	1	2
1010	5	6
1012	6	7
1013	3	4
1015	2	3
1015	4	5

Query with Avg (Input):

```
SELECT AVG(QUANTITY)

FROM CONSUMES_2023;
```

Query with Avg (Output):

AVG(QUANTITY)
4.5

Query with Compound Conditions (Input):

```
SELECT PATIENT_NO, PATIENT_NAME, PHYSICIAN_ID, BED_NO

FROM PATIENT_2023

WHERE PHYSICIAN_ID = 1235 and BED_NO= 0004;
```

Query with Compound Conditions (Output):

PATIENT_NO	PATIENT_NAME	PHYSICIAN_ID	BED_NO
1013	Julie Su	1235	4

Ny'Chadrialynne's Question:

Question:

Would Mountain View Community Hospital benefit from using these techniques to navigate and manage their databases?

Answer:

Yes! Mountain View Community Hospital or any hospital would benefit from these databases tremendously. They can use these databases to advance their techniques in technology and manage the specific types of data in their expansive database.

McKenzie's Question:

Why is query filtering important and provide an example.

- There can be multiple levels of sorting that can be done by the primary sort key and secondary sort key.
- SELECT * FROM PATIENT_2023 WHERE PHYSICIAN_ID>1234

  ORDER BY BED_NO,PATIENT_NAME;

Romello's Question:

What are the benefits of using in between conditions? Give an example with explanation.

A. The BETWEEN conditions allows an alternative to using two comparison operators. It specifies a range of values within a condition. The following example shows PATIENT_NO, PATIENT_NAME, PHYSICIAN_ID,BED_NO from PATIENT_2023 where the PHYSICIAN_ID is 1234, and BED_NO is BETWEEN 0001-0005.

```
SELECT PATIENT_NO, PATIENT_NAME, PHYSICIAN_ID, BED_NO
FROM PATIENT_2023
WHERE PHYSICIAN_ID = '1234'
AND BED_NO BETWEEN '0001' AND '0005';
```