

SQL Unit 2

Simple SQL Queries with Selection and Projection

Ryan Nixon

First Off

- Lab day assignments, groups
- Assn 1, Q9
- Remember, assignments are your study guide

SQL Queries

- Text used to request information from or submit information to a database
- Databases need to know what you're asking
 - What information? (Columns & conditions)
 - Where is it stored? (Table name)

Selecting Information

- Data is requested with a “SELECT” statement:
 - `SELECT *` selects all columns
 - `SELECT id, first, last` selects only the primary key and the columns named “first” and “last”
 - `SELECT first AS firstname` selects the “first” column and names it “firstname”

Selecting Information

- The order of columns does not matter.
Repeating the same column is also allowed,
and produces the same result
- The practice of choosing which columns to
display is known as “projection”

Selecting Information

- “FROM” is used to specify where to pull the data from:
 - `FROM student` targets the student table
 - `FROM student AS s` targets the student table, but uses “s” as an alias to the table. This is useful when pulling from the same table twice

Some Examples

```
SELECT *  
FROM grade;
```

```
SELECT id, student_id, assn_id, points  
FROM grade;
```

```
SELECT id, first AS 'First Name', last AS last_name, age  
FROM person
```

```
SELECT id, model_id, vin, lic AS [License Number]  
FROM car;
```

```
SELECT id, artist_id, album_id, pos AS position, length  
FROM track AS song
```

Note the semicolon

- Semicolons are used to separate multiple statements in a single “transaction”
- A transaction is any collection of multiple statements, whether they be SELECTs or other queries
- Later in the semester we will see how we can control how these transactions function

Query Results

```
SELECT *  
FROM grade
```

```
SELECT id, student_id, assn_id, points  
FROM grade
```

id	student_id	assn_id	points
1	1	1	5.00
2	6	3	27.00
3	6	1	9.00
4	5	2	13.50
5	3	3	25.00

```
SELECT id, first AS 'First Name', last AS last_name, age  
FROM person
```

id	First Name	last_name	age
1	Jacob	Williams	37
2	Emma	Emmerich	25
3	Nathan	Drake	32
4	Albert	Wesker	41
5	Alexandra	Vance	29
6	Christopher	Garden	46

Ordering Queries

- “ORDER BY” is placed at the end of a query to order its results
 - `ORDER BY age` sorts the results by age, ascending
 - `ORDER BY points DESC` sorts the results by points, descending
 - `ORDER BY age, points` first sorts by age, then by points if necessary

Query Results

```
SELECT id, student_id, assn_id, points
FROM grade
ORDER BY assn_id, points
```

id	student_id	assn_id	points
1	1	1	5.00
3	6	1	9.00
4	5	2	13.50
5	3	3	25.00
2	6	3	27.00

```
SELECT id, first AS 'First Name', last AS last_name, age
FROM person
ORDER BY age DESC
```

id	First Name	last_name	age
6	Christopher	Garden	46
4	Albert	Wesker	41
1	Jacob	Williams	37
3	Nathan	Drake	32
5	Alexandra	Vance	29
2	Emma	Emmerich	25

See it in Action

Filtering Queries

- Filtering the displayed rows is known as “selection”
- “WHERE” is used to apply filters
- Where clauses, or conditions, use boolean logic to determine what to display

Boolean Expressions

- Commonly used in logic. Boolean (True or False) expressions can be used to explicitly define an argument
- “If this is True, then that must be False”
- “If all of these are True then this must be True as well”
- Follows order of operations similar to mathematics

Boolean Expressions

- $>$ Greater than
- $<$ Less than
- \geq Greater than/equal to
- \leq Less than/equal to
- $=$ Equal To
- \neq Not equal to
- This AND That
Both This and That must be true
- This OR That
Either This or That can be true
- NOT That
That must not be true

Boolean Expressions

- Expressions can be chained and preference set using parentheses
- `(This >= That OR That > This) AND That != 5 AND Those = 'String'`
- Note that the parentheses always equates to True, but due to the AND components the other items must also be True

Data Types in Expressions

- Note the `'String'` in the expression. This is how strings appear in SQL and how they must be used in WHERE clauses.
- To escape (ignore) a quotation mark use two of them, such as `'It''s a wonderful day'`.
- Dates & times also appear in quotations
(`'2012-08-29'`)
- Numerics should not appear in quotations
(`This = 5`) **or** (`This = 5.0`)

Data Types in Expressions

- NULL is a special data type representing nothing
- To query for NULL, use `IS NULL` or `IS NOT NULL`
- ```
SELECT *
FROM person
WHERE date_of_death IS NOT NULL
```

# Data Types in Expressions

- “LIKE” can be used when dealing with partial strings
- % (SQL) or \* (MS Access) are used as wildcards. Any series of characters will match this
- ? (MS Access) represents a single character
- # (MS Access) represents a single number/digit

# Data Types in Expressions

- All tracks that begin with “Black”:
- ```
SELECT *  
FROM track  
WHERE title LIKE 'Black%'
```
- All essay papers containing some common typos:
- ```
SELECT *
FROM essay
WHERE content LIKE '%distracti0n%' OR
content LIKE '%cappuccino%'
```

# Eliminating Duplicates

- Queries can often produce duplicate results. In this case it is best to use **DUPLICATE** in your select statement
- This removes all duplicate rows from the result set, but not duplicate columns
- `SELECT DISTINCT firstname, lastname  
FROM person`
- Note the exclusion of the primary key

# See it in Action

# Notes

- Assn 2 up tonight, due 9/24
- Lab hours on Wednesday in SSB 172