Gerard Anthony Resaba 3/BSCS/A

Application Development

**Basics**

- USE sql\_store; SELECT \*

- FROM customers

- WHERE state = ‘CA’

- ORDER BY first\_name

- LIMIT 3;

- (SQL is not a case-sensitive language. In MySQL, every statement must be terminated with a semicolon.)

**Comments**

- use comments to add notes to your code

**Select Clause**

**-** SELECT (points \* 10 + 20) AS discount\_factor

- FROM customers

Order of operations:

- Parenthesis

- Multiplication/Division

- Addition/Subtraction

Removing duplicates

- SELECT DISTINCT state

- FROM customers

**WHERE Clause**

Use the where clause to filter data.

Comparison operators:

- Greater than ( > )

- Greater than or equal to: ( > = )

- Less than: ( < )

- Less than or equal to: ( < = )

- Equal: ( = )

- Not equal: ( < > )

- Not equal: ( != )

**Logical Operators**

AND (both conditions must be True)

- SELECT \*

- FROM customers

- WHERE birthdate > ‘1990-01-01’ AND points > 1000

OR (at least one condition must be True)

- SELECT \*

- FROM customers

- WHERE birthdate > ‘1990-01-01’ OR points > 1000

NOT (to negate a condition)

- SELECT \*

- FROM customers

- WHERE NOT (birthdate > ‘1990-01-01’)

**IN Operator**

Returns customers in any of these states: VA, NY, CA

- SELECT \*

- FROM customers

- WHERE state IN (‘VA’, ‘NY’, ‘CA’)

**BETWEEN Operator**

- SELECT \*

- FROM customers

- WHERE points BETWEEN 100 AND 200

**LIKE Operator**

Returns customers whose first name starts with b

- SELECT \*

- FROM customers

- WHERE first\_name LIKE ‘b%’

- %: any number of characters

- \_: exactly one character

**REGEXP Operator**

Returns customers whose first name starts with a

- SELECT \*

- FROM customers

- WHERE first\_name REGEXP ‘^a’

- ^: beginning of a string

- $: end of a string

- |: logical OR

- [abc]: match any single characters

- [a-d]: any characters from a to d

**IS NULL Operator**

Returns customers who don’t have a phone number

- SELECT \*

- FROM customers

- WHERE phone IS NULL

**ORDER BY Clause**

Sort customers by state (in ascending order), and then

by their first name (in ascending order)

- SELECT \*

- FROM customers

- ORDER BY state, first\_name DESC

**LIMIT Clause**

Return only 3 customers

- SELECT \*

- FROM customers

- LIMIT 3

Skip 6 customers and return 3

- SELECT \*

- FROM customers

- LIMIT 6, 3

**Inner Joins**

- SELECT \*

- FROM customers c

- JOIN orders o ON c.customer\_id = o.customer\_id

**Outer Joins**

Returns all customers whether they have any order or not

- SELECT \*

- FROM customers c

- LEFT JOIN orders o ON c.customer\_id = o.customer\_id

**USING Clause**

If column names are exactly the same, you can simplify the join with the USING clause.

- SELECT \*

- FROM customers c

- JOIN orders o

- USING (customer\_id)

**Cross Joins**

Combine every color with every size

- SELECT \*

- FROM colors

- CROSS JOIN sizes

**Unions**

Combine records from multiple result sets

- SELECT name, address

- FROM customers

- UNION

- SELECT name, address

- FROM clients

**Inserting Data**

Insert a single record

- INSERT INTO customers(first\_name, phone, points)

- VALUES (‘Adi’, NULL, DEFAULT)

Insert multiple single records

- INSERT INTO customers(first\_name, phone, points)

- VALUES (‘Adi’, NULL, DEFAULT), (‘Gerard’, ‘1234’, 10)