Set #14: More SQL SELECT, DDL and DML

**SELECT Summary**

- `SELECT [DISTINCT] column_name(s) | aggregate_expr`  
  FROM `table_name(s)`  
  WHERE `row_conditions`  
  GROUP BY `grouping_columns`  
  HAVING `group_conditions`  
  ORDER BY `column_name(s)` [ASC/DESC]
Limit (MySQL specific)

- Limit the number of rows in the result
- SELECT \[DISTINCT\] column_name(s) | aggregate_expr
  FROM table_name(s)
  WHERE row_conditions
  GROUP BY grouping_columns
  HAVING group_conditions
  ORDER BY column_name(s) [ASC/DESC]
  LIMIT [offset,] row_count
- Returns at most row_count rows, starting with offset
  (offset of first row is 0)

UNION

- Students(Alpha, Lname, FName, Email)
- GraduatedStudents(Alpha2, Lname2, FName2, Email2)
- List the Alpha, last name and first name of all students
  (current or graduated)

SELECT Alpha, Lname, FName
FROM Students
UNION
SELECT Alpha2, Lname2, FName2
FROM GraduatedStudents

• Same number of columns and data types in the 2 select
• Duplicates in results are eliminated
  by default (use UNION ALL if want to keep duplicates)
SQL DDL and DML

- SQL statements can be divided into two categories:
  - **Data definition language (DDL)** statements
    - Used for creating and modifying tables, views, and other structures
    - CREATE, DROP, ALTER
  - **Data manipulation language (DML)** statements.
    - Used for queries and data modification
    - INSERT, DELETE, UPDATE, SELECT

SQL DML

- **Data manipulation language (DML)** statements.
  - Used for queries and data modification
  - INSERT
  - DELETE
  - UPDATE
  - SELECT
INSERT Statement

INSERT INTO table_name (column_list) VALUES (data_values)
INSERT INTO table_name (column_list) select_statement

INSERT command:

INSERT INTO Students (StudentNumber, StudentLastName, StudentFirstName)
VALUES (190, 'Smith', 'John'); -- this inserts nulls/defaults in the columns not named

INSERT INTO Students VALUES(190, 'Smith', 'John', 'jsmith@usna.edu', '410-431-3456')
-- this expects all columns to have values in the value list; use null if value is null

- Bulk INSERT:
  INSERT INTO Students (StudentNumber, StudentLastName, StudentFirstName, Email, PhoneNumber)
  SELECT *
  FROM Second_Class_Students;

UPDATE Statement

UPDATE table_name
SET column_name1 = expression1 [,column_name2 = expression2,... ]
[ WHERE search_condition ]

- UPDATE command:
  UPDATE Students
  SET PhoneNumber = '410-123-4567'
  WHERE StudentNumber = 673;

- BULK UPDATE command:
  UPDATE Students
  SET PhoneNumber = '410-123-4567'
  WHERE StudentLastName = 'Doe';

<table>
<thead>
<tr>
<th>Student Number</th>
<th>Student LastName</th>
<th>Student FirstName</th>
<th>Email</th>
<th>PhoneNumber</th>
</tr>
</thead>
<tbody>
<tr>
<td>190</td>
<td>Smith</td>
<td>John</td>
<td><a href="mailto:jsmith@usna.edu">jsmith@usna.edu</a></td>
<td>410-431-3456</td>
</tr>
<tr>
<td>673</td>
<td>Doe</td>
<td>Jane</td>
<td><a href="mailto:jdoe@usna.edu">jdoe@usna.edu</a></td>
<td></td>
</tr>
<tr>
<td>312</td>
<td>Doe</td>
<td>Bob</td>
<td><a href="mailto:bred@usna.edu">bred@usna.edu</a></td>
<td>443-451-7865</td>
</tr>
</tbody>
</table>
DELETE Statement

DELETE FROM `table_name`
[WHERE `search_condition`]

- DELETE command:
  ```sql
  DELETE FROM Students
  WHERE StudentNumber = 190;
  ```
  If you omit the WHERE clause, you will delete every row in the table!!!

- Another example:
  ```sql
  DELETE FROM Departments
  WHERE DepartmentName = 'ComSci'
  ```

Integrity constraints?!

- If Foreign Key constraint in Students referencing Departments:
  - if ON DELETE No ACTION, department cannot be deleted as long as there are students in that department
  - If ON DELETE CASCADE, all students from a department are deleted when department is deleted

SQL DDL

- Data definition language (DDL) statements.
  - Used for creating and modifying tables, views, and other structures
  - CREATE
  - ALTER
  - DROP
Creating Tables Review

CREATE TABLE table_name(
  column_name1 column_type1 [constraints1],
  ...
) [
[CONSTRAINT constraint_name] table_constraint
)

Table constraints:
- NULL/NOT NULL
- PRIMARY KEY (columns)
- UNIQUE (columns)
- CHECK (conditions)
- FOREIGN KEY (local columns) REFERENCES foreign_table
  (foreign_columns) [ON DELETE action_d ON UPDATE action_u]

Specify surrogate key in MySQL:
  column_name int_type AUTO_INCREMENT

Modifying Tables

- ALTER TABLE table_name clause

Clauses: – some are DBMS specific!
  ADD COLUMN column_name column_type [constraints]
  DROP COLUMN column_name
  MODIFY COLUMN column_name column_type [constraints]
  ADD CONSTRAINT constraint
  DROP PRIMARY KEY/FOREIGN KEY constraint_name
ALTER TABLE Examples

- ALTER TABLE Students ADD COLUMN BirthDate datetime NULL
- ALTER TABLE Students DROP COLUMN BirthDate
- ALTER TABLE Student ADD CONSTRAINT FK_Department
  FOREIGN KEY (MajorDepartmentName)
  REFERENCES Departments (DepartmentName)
  ON DELETE NO ACTION
  ON UPDATE CASCADE

Removing Tables

- DROP TABLE table_name
  
  DROP TABLE Departments;

- If there are constraints dependent on table:
  - Remove constraints
  - Drop table
  
  ALTER TABLE Students
  DROP FOREIGN KEY FK_Department;

  DROP TABLE Departments;
Summary: SQL DDL and DML

- **Data definition language (DDL) statements**
  - Used for creating and modifying tables, views, and other structures
  - CREATE, ALTER, DROP

- **Data manipulation language (DML) statements.**
  - Used for queries and data modification
  - INSERT, DELETE, UPDATE, SELECT