SY306 Web and Databases for Cyber Operations
Set #13: Databases - The Relational Model and SQL

https://www.w3schools.com/sql/default.asp
https://www.tutorialspoint.com/sql/
Relational Model - Tables

<table>
<thead>
<tr>
<th>UserName</th>
<th>Gender</th>
<th>Age</th>
<th>Email</th>
<th>Passwd</th>
</tr>
</thead>
<tbody>
<tr>
<td>Alice</td>
<td>F</td>
<td>19</td>
<td><a href="mailto:a@usna.edu">a@usna.edu</a></td>
<td>wad354daa</td>
</tr>
<tr>
<td>Bob</td>
<td>M</td>
<td>21</td>
<td><a href="mailto:bob@usna.edu">bob@usna.edu</a></td>
<td>Ddadha21hda</td>
</tr>
<tr>
<td>Greg</td>
<td>M</td>
<td>19</td>
<td><a href="mailto:greg@usna.edu">greg@usna.edu</a></td>
<td>Sada3145ada</td>
</tr>
<tr>
<td>Jane</td>
<td>F</td>
<td>18</td>
<td><a href="mailto:jane@gmail.com">jane@gmail.com</a></td>
<td>sadad45da45</td>
</tr>
</tbody>
</table>

Keys

- Key
- Primary key
- Candidate key
- Surrogate key
Foreign keys

ICE: Is This a Relation? Why?

<table>
<thead>
<tr>
<th>A</th>
<th>X</th>
<th>C</th>
<th>A</th>
</tr>
</thead>
<tbody>
<tr>
<td>John</td>
<td>Ryan</td>
<td>MD</td>
<td><a href="mailto:jr@gmail.com">jr@gmail.com</a></td>
</tr>
<tr>
<td>Bob</td>
<td>Smith</td>
<td>MD, VA, NY</td>
<td><a href="mailto:bsm@gmail.com">bsm@gmail.com</a></td>
</tr>
<tr>
<td>Alice</td>
<td>Brown</td>
<td>CA</td>
<td></td>
</tr>
<tr>
<td>Jane</td>
<td>Doe</td>
<td>WA</td>
<td><a href="mailto:jd@yahoo.com">jd@yahoo.com</a></td>
</tr>
<tr>
<td>John</td>
<td>Ryan</td>
<td>MD</td>
<td><a href="mailto:jr@gmail.com">jr@gmail.com</a></td>
</tr>
<tr>
<td>5</td>
<td>4</td>
<td>5</td>
<td>4</td>
</tr>
</tbody>
</table>
Structured Query Language

• **Data definition language (DDL)**
  – CREATE, DROP, ALTER

• **Data manipulation language (DML)**
  – INSERT, DELETE, UPDATE, SELECT

Creating Tables

```sql
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]

**action_d:** NO ACTION|CASCADE|SET NULL|SET DEFAULT

Specify surrogate key in MySQL:
```
column_name int_type AUTO_INCREMENT
```
CREATE TABLE Example

- Courses(\textit{Cid}, Cname, Cdept, CreditHours)

ICE: Write the SQL to create the Enrolled table

- Enrolled(\textit{Alpha}, \textit{Cid}, Semester, Grade)
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
Removing Tables

- DROP TABLE *table_name*

  ```sql
  DROP TABLE Departments;
  ```

- If there are constraints dependent on table:
  - Remove constraints
  - Drop table

  ```sql
  ALTER TABLE Students
  DROP FOREIGN KEY FK_Department;
  ```

  ```sql
  DROP TABLE Departments;
  ```

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)

**UPDATE Statement**

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

**INSERT command:**

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

**UPDATE command:**

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

- BULK 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:
  DELETE FROM Students
  WHERE StudentNumber = 190;

If you omit the WHERE clause, you will delete every row in the table!!!

- Another example:
  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

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