SQL Basics

Dr. Isaac Griffith Idaho State University

Storing Data

Idaho State University | Computer Science

- It is inevitable that we will need to persist data between executions of our application
- In 2263, we utilized a naive approach for doing this via JSON and Serialization
- However, often we need something far more robust…

Outcomes

Idaho State University | Computer Science

After today's lecture you will be able to:

- Identify different Database Management Systems
- Transform a basic UML Class Diagram into a database schema and tables
- Utilize fundamental knowledge of SQL Data Definition Language to construct both
  - Database schema
  - Database tables
- Utilize fundamental knowledge of SQL Data Manipulation Language to
  - Inserting, Updating, and Deleting Data

DBMS

CS 3321

DBMS

- Database Management System (DBMS) – provide a system for optimally storing and retrieving data.

- There are essentially three main types:
  - **Relational** – MySQL, MariaDB, Postgres, SQLite, RDS, and many others
  - Non-Relational (i.e., No-SQL) – Amazon DynamoDB, MongoDB, Big Table
  - Others Databases
    - Graph Databases
    - Time Series Data Stores
    - Object Data Stores

SQL

CS 3321

SQL

- SQL is a data manipulation language
- SQL is not a programming language
- SQL commands are interpreted by the DBMS engine
- SQL commands can be used interactively as a query language with in the DBMS
- SQL commands can be embedded within programming languages

3 Types of SQL Commands

Idaho State University | Computer Science

- **Data Definition Language (DDL):**
  - Commands that define a database - `Create`, `Alter`, `Drop`
- **Data Manipulation Language (DML):**
  - Commands that maintain and query a database
- Data Control Language (DCL)
  - Commands that control a database, including administering privileges and committing data.

Data Definition Language

CS 3321

Some SQL Data Types

Idaho State University | Computer Science

MySQL and MS SQL Server

- String types:
  - `CHAR(n)` – fixed-length character data, `n` characters long, max length is 255 characters
  - `VARCHAR(n)` – variable length character data, max length is 65535 characters
  - `TEXT(n)` – String data with max size of 65,535 bytes
  - `LONGTEXT` – String data with a max size of 4,294967,295 characters
- Numeric Types
  - `BOOL` or `BOOLEAN` – Zero is false, all other values are true
  - `INT(n)` or `INTEGER(n)` – A medium sized integer
  - `FLOAT(p)` – decimal integer with `p` numbers after the decimal point
- Date/Time Type:
  - `DATE` – fixed-length date/time in YYYY-MM-DD format

Some SQL Data Types

SQLite

- String types:
  - `TEXT` – stores string data
  - `BLOB` – stores other data
- Numeric types:
  - `INTEGER` – stores signed integer data
  - `REAL` – stores floating point data
- Others
  - There is no specific datatype for **boolean** or **dates**
  - Booleans are typical stored as `INTEGER`
  - Dates can be stored as `INTEGER`, `REAL`, or `TEXT` values

SQL Database Definition

Idaho State University | Computer Science

- Data Definition Languae (DDL)
- Major `CREATE` statements:
  - `CREATE SCHEMA` – defines a portion of the database owned by a particular user
  - `CREATE TABLE` – defines a table and its columns
  - `CREATE VIEW` – defines a logical table from one or more views
- `ALTER` statements

Table Creation

Idaho State University | Computer Science

General Syntax

```
CREATE TABLE tablename (
  {column definition [table constraint]},
);
```

Where column definitions:

```
column_name datatype[(size)] [column_constraint]
       [default value] [collate clause]
```

And table constraints:

```
[CONSTRAINT constraint_name] Constraint_type
       [constraint_attrs]
```

Steps in table creation:

1. Identify data types for attributes
2. Identify columns that can and cannot be null
3. Identify columns that must be unique (candidate keys)
4. Identify primary key-foreign key mates
5. Determine default values
6. Identify constraints on columns (domain specifications)
7. Create the table and associated indexes

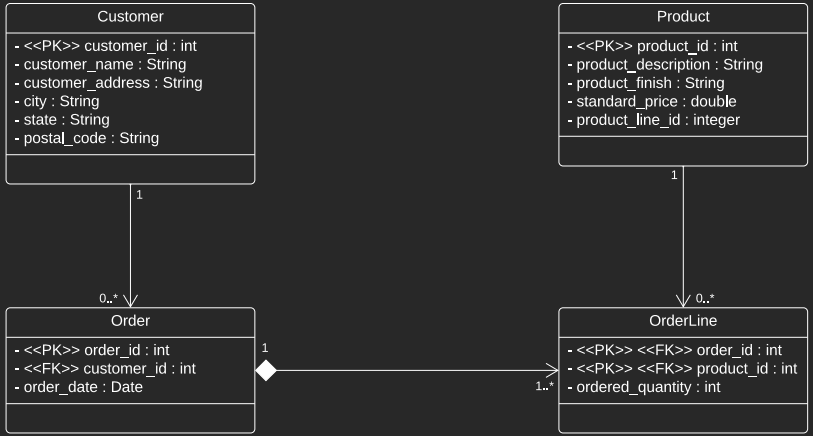
Modeling Databases

Idaho State University | Computer Science

- There are several methods to model a database
  - ER Diagrams
  - EER Diagrams
  - **UML Class Diagrams**

- Given that we are discussing the backend data of a software system and later we will be working with ORM, we will use **UML Class Diagrams**
  - Each class is a table, and each object stored is a row
  - Fields represent columns of the table
  - Operations, do not have a place here
  - **Note:** be careful concerning circular dependencies

Example Model



Customer Table

Idaho State University | Computer Science

SQLite

```
CREATE TABLE customers (
  customer_id       INTEGER NOT NULL PRIMARY KEY AutoIncrement,
  customer_name     VARCHAR NOT NULL,
  customer_address  VARCHAR,
  city              VARCHAR,
  state             VARCHAR,
  postal_code       VARCHAR
);
```

MySQL

```
CREATE TABLE customers (
  customer_id       INTEGER NOT NULL PRIMARY KEY Auto_Increment,
  customer_name     VARCHAR(25) NOT NULL,
  customer_address  VARCHAR(30),
  city              VARCHAR(20),
  state             VARCHAR(2),
  postal_code       VARCHAR(9)
);
```

Order Table

Idaho State University | Computer Science

SQLite

```
CREATE TABLE orders (
  order_id     INTEGER NOT NULL PRIMARY KEY AutoIncrement,
  order_date   NUMERIC NOT NULL,
  customer_id  INTEGER REFERENCES customers (customer_id)
);
```

MySQL

```
CREATE TABLE orders (
  order_id     INTEGER NOT NULL PRIMARY KEY AutoIncrement,
  order_date   DATETIME NOT NULL,
  customer_id  INTEGER REFERENCES customers (customer_id)
);
```

Product Table

Idaho State University | Computer Science

SQLite

```
CREATE TABLE products (
  product_id           INTEGER NOT NULL PRIMARY KEY AutoIncrement,
  product_description  VARCHAR,
  product_finish       VARCHAR,
  standard_price       REAL,
  product_line_id      INTEGER
);
```

MySQL

```
CREATE TABLE products (
  product_id           INTEGER NOT NULL PRIMARY KEY AutoIncrement,
  product_description  VARCHAR(50),
  product_finish       VARCHAR(20),
  standard_price       DECIMAL(6,2),
  product_line_id      INTEGER
);
```

OrderLine Table

Idaho State University | Computer Science

SQLite

```
CREATE TABLE order_lines (
  order_id          INTEGER NOT NULL REFERENCES orders (order_id),
  product_id        INTEGER NOT NULL REFERENCES products (product_id),
  ordered_quantity  INTEGER,
  CONSTRAINT ORDER_LINE_PK PRIMARY KEY (order_id, product_id)
);
```

MySQL

```
CREATE TABLE order_lines (
  order_id          INTEGER NOT NULL REFERENCES orders (order_id),
  product_id        INTEGER NOT NULL REFERENCES products (product_id),
  ordered_quantity  INTEGER,
  CONSTRAINT ORDER_LINE_PK PRIMARY KEY (order_id, product_id)
);
```

Changing and Removing Tables

Idaho State University | Computer Science

- `ALTER TABLE` statement allows you to change column specifications:

```
ALTER TABLE customers ADD (TYPE VARCHAR(2));
```

- `DROP TABLE` statement allows you to remove tables from your schema:

```
DROP TABLE customers;
```

Schema Definition

- Control processing/storage efficiency
  - Choice of indexes
  - File organizations for base tables
  - File organizations for indexes
  - Data clustering
  - Statistics maintenance
- Creating indexes
  - Speed up random/sequential access to base table data
  - Example

Data Manipulation Language

CS 3321

Data Manipulation Language (DML)

Idaho State University | Computer Science

Four Basic Commands:

- `INSERT`
- `UPDATE`
- `DELETE`
- `SELECT`

Inserting Data

Idaho State University | Computer Science

- Puts **ONE** row into a table
- Column list is optional if you plan to insert a value into every column and in the same order as the table
- If you wish to change the order of data, the column list is needed
  - value list must match
  - you can use `NULL` or blank values
- Columns left out, will have a value of `NULL`
  - only if the column is able to be `NULL`

Syntax:

```
INSERT INTO tablename (column-list)
VALUES (value-list);
```

Example:

```
INSERT INTO course (course_code, course_name,
  credit_hours)
VALUES ('MIS499', 'ADVANCED ORACLE', 4);
```

Deleting Data

Idaho State University | Computer Science

- Removes rows from a table

- Delete certain rows:

```
DELETE FROM customer_t WHERE
STATE = 'HI';
```

- Delete all rows

```
DELETE FROM customer_t;
```

Updating Data

Idaho State University | Computer Science

- Modifies data in existing rows

```
UPDATE product_t SET unit_price =
775 WHERE product_id = 7;
```

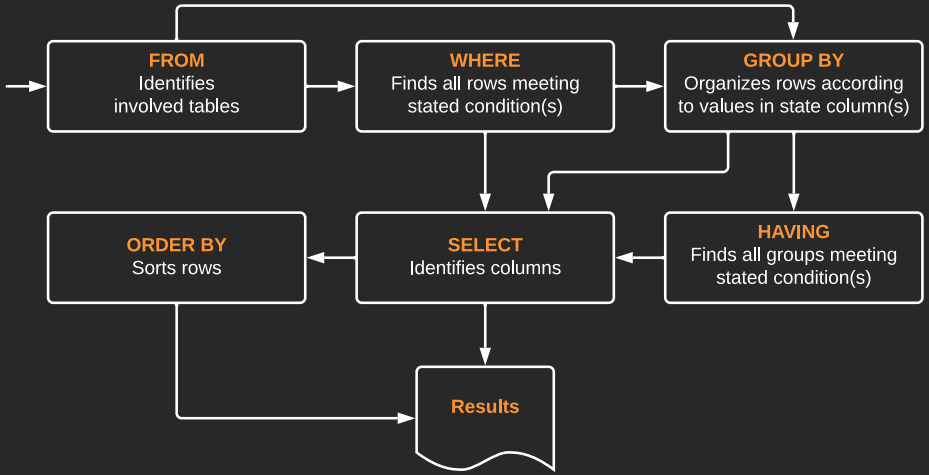
Retrieving Data

Idaho State University | Computer Science

- We use the `SELECT` statement for queries on a single or multiple tables

- Clauses of the `SELECT` statement:
  - `SELECT` – List the columns (and expressions) that should be returned from the query
  - `FROM` – Indicates the table(s) or view(s) from which data will be obtained
  - `WHERE` – Indicate the conditions under which a row will be included in the result
  - `GROUP BY` – Indicate categorization of results
  - `HAVING` – Indicate the conditions under which a category (group) will be included
  - `ORDER BY` – Sorts the result according to specified criteria

Query Processing



SELECT Example

Idaho State University | Computer Science

- Find products with standard price less than $275

```
SELECT product_name, standard_price
FROM products
WHERE standard_price < 275;
```

- Aliases allow for alternative column or table names

```
SELECT cust.customer as NAME,
  cust.customer_address
FROM customers cust
WHERE name = 'Home Furnishings';
```

Counting Rows

Idaho State University | Computer Science

- You can use the `COUNT` **aggregate function** to find totals

```
SELECT count(*) FROM order_lines
WHERE order_id = 1004;
```

- **Note:** with aggregate functions you can't have single-valued columns included in the select clause

Boolean Operators

Idaho State University | Computer Science

- `AND`, `OR`, and `NOT` operators can be used to customize conditions in a `WHERE` clause

```
SELECT product_description, product_finish,
  standard_price
FROM products
WHERE (product_description LIKE '%Desk'
OR product_description LIKE '%Table')
AND unit_price > 300;
```

Order By

- Example: Sort the results first by STATE, and within a state by CUSTOMER_NAME

```
SELECT customer_name, city, state
FROM customers
WHERE state IN ('FL', 'TX', 'CA', 'HI')
ORDER BY state, customer_name;
```

Group By

Idaho State University | Computer Science

- Group by is for use with aggregate functions
  - **Scalar aggregate:** single value returned from SQL query with aggregate function
  - **Vector aggregate:** multiple values returned from SQL query with aggregate function (via `GROUP BY`)

```
SELECT state, count(state)
FROM customers
GROUP BY state;
```

- **Note:** you can use single-value fields with aggregate functions if they are included in the `GROUP BY` clause

The HAVING Clause

- We can qualify results by categories using the HAVING clause when using GROUP BY

```
SELECT state, count(state)
FROM customers
GROUP BY state
HAVING count(state) > 1;
```

For Next Time

- Review the Reading
- Review this Lecture
- Come to Class

Are there any questions?