Lecture 9:

- Creating Tables in SQL (cont)

Announcements:

- HW 3 out, due Tues
- Note: Project part 1 is due next Thursday (10/5)
- Quiz 4 on Tues (SQL basics: creating and populating tables)
Adding Foreign Keys:

Adding an account table:

    account(acct_id, acct_name, main_branch)

    ... where main_branch is a FK to the branch table

CREATE TABLE account (  
    acct_id INT UNSIGNED,  
    acct_name TINYTEXT,  
    main_branch VARCHAR(50),  
    PRIMARY KEY (acct_id),  
    FOREIGN KEY (main_branch) REFERENCES branch (branch_name)  
);  

In general, FKs take the form:

    FOREIGN KEY (att₁, ..., attₙ) REFERENCES table (att₁, ..., attₙ)

Note: Foreign keys impact order tables can be dropped!

- must drop table with foreign key, before dropping referenced table
- in general, can drop tables in reverse creation order
Inserting Rows

Basic forms of row insertion ...

- \texttt{INSERT INTO table VALUES (v1, v2, ...);}
- \texttt{INSERT INTO table VALUES (v1, v2, ...), (v3, v4, ...), ...;}
- \texttt{INSERT INTO table(a1, a2) VALUES (v1, v2);}
- \texttt{INSERT INTO table SET a1 = v1, a2 = v2, ...;}

For example:

- \texttt{INSERT INTO account VALUES (101, 'Alice', 'Central');}
- \texttt{INSERT INTO account VALUES (102, 'Bob', 'Central'), (103, 'Charlie', 'Shadle');}

To View Table Rows

Can use the following SQL Query (more later)

- \texttt{SELECT * FROM table;}
  \(\ldots\) \textit{note: SELECT \(\approx\) \(\pi\)}

For example:

- \texttt{SELECT *}
  \texttt{FROM account;}
- \texttt{SELECT *}
  \texttt{FROM branch;}

**Additional Table Constraints**

**SQL NOT NULL constraints:** states the attribute is *required* (vs optional)

```sql
CREATE TABLE account (
    acct_id INT UNSIGNED NOT NULL, -- Redundant with PK
    acct_name TINYTEXT NOT NULL, -- NULL not allowed
    main_branch VARCHAR(50) NOT NULL, -- must have branch
    PRIMARY KEY (acct_id),
    FOREIGN KEY (main_branch) REFERENCES branch (branch_name)
);
```

**SQL UNIQUE constraints:** for enforcing non-primary candidate keys

```sql
CREATE TABLE branch (
    branch_name VARCHAR(50),
    address TINYTEXT NOT NULL,
    phone VARCHAR(12) NOT NULL,
    PRIMARY KEY (branch_name),
    UNIQUE (phone) -- phone also identifies branch
);
```
SQL **CHECK constraints**: enforces conditional expressions

```sql
CREATE TABLE item_loan (  
    acct_id INT UNSIGNED NOT NULL,  
    barcode INT UNSIGNED NOT NULL,  
    checkout_date DATE NOT NULL,  
    due_date DATE NOT NULL,  
    return_date DATE,  
    PRIMARY KEY (acct_id, barcode, checkout_date),  
    FOREIGN KEY (acct_id) REFERENCES account (acct_id),

    CONSTRAINT valid_barcode_value CHECK  
    (barcode > 0 and barcode <= 1000000),

    CONSTRAINT valid_due_date CHECK  
    (checkout_date < due_date),

    CONSTRAINT valid_return_date  
    CHECK (return_date IS NULL or return_date >= checkout_date)  
);
```

Can also "inline" the CHECK constraint

Can also name FK constraints (can be good for updating/removing later)

```sql
    CONSTRAINT item_loan_acct_id_fk  
    FOREIGN KEY (acct_id) REFERENCES account (acct_id),
```

Note that if you receive a warning, you can view it in MariaDB using:

```sql
    >> SHOWN WARNING;
```
Modifying Rows

Removing all rows from tables (without removing the entire table):

    DELETE FROM account;

Removing specific rows using query conditions (more later):

    DELETE FROM account
    WHERE acct_id = 101;

Updating attribute value of all rows

    UPDATE loan SET return_date = '2021-10-12';

Updating attribute value of specific rows

    UPDATE loan SET return_date = '2021-10-12' WHERE acct_id = 101;
Modifying Table Schemas

To drop a column from an existing table ...

    ALTER TABLE my_table DROP COLUMN my_column;

To change name and type of an attribute ...

    ALTER TABLE my_table CHANGE my_column my_new_column data_type;

To change type of an attribute (leave name unchanged) ...

    ALTER TABLE my_table MODIFY my_column new_data_type;

To add a new attribute ...

    ALTER TABLE my_table ADD my_new_column data_type;

Can also ADD/DROP named constraints ...

    ALTER TABLE my_table ADD CONSTRAINT my_table_fk ...;
    ALTER TABLE my_table DROP CONSTRAINT my_table_fk;