- Primary and Foreign Keys
    - Key and Index
        - KEY – identifies row in table
        - INDEX – ordered collection of keys
        - UNIQUE – every row in different key
            - Or every entry in index only appears once
    - Primary Key
        - A column (or column set) that uniquely identifies a row within table
        - Must be unique, and not NULL
        - One primary key for each table
        - Can add while creating table or after
            - Ex: ALTER TABLE table
                ADD PRIMARY KEY (key);
    - Foreign Key
        - A key column (or column set) in the child table that matches the primary key in the parent table
        - NULL foreign keys allowed
    - Adding Foreign Key Example
        - ALTER TABLE people
        - ADD CONSTRAINT FK_PeopleAddress
        - FOREIGN KEY (address_id) REFERENCES addresses(id);
    - Dropping Foreign Key Example
        - ALTER TABLE people
        - DROP FOREIGN KEY FK_PeopleAddress,
        - DROP INDEX FK_PeopleAddress;
- Constraints
    - Add Constraint Example
        - ALTER TABLE table
        - ADD CONSTRAINT constraint UNIQUE (column_name);
    - Drop Constraint Example
        - ALTER TABLE table
        - DROP INDEX constraint;
    - Constraint Types
        - NOT NULL
        - UNIQUE
        - PRIMARY KEY
        - FOREIGN KEY
        - CHECK
            - Controls values that can be inserted into a column
        - DEFAULT
            - Can set default value if no value inserted into column

- Aggregate Functions
  - *Perform calculations on data within a column, and return one result row*
  - count
    - Ex: SELECT count(*) FROM customers;
    - Ex: SELECT count(first_name) FROM customers
    - WHERE last_name = 'Smith';
    - *Does not include* NULL *values when counting on specific column*
  - Sum
    - Ex: SELECT sum(num_seats) FROM rooms;
    - *Can only sum values of numeric columns*
  - min/max
    - Ex: SELECT max(length_min) FROM films;
  - avg (mean)
    - Ex: SELECT avg(length_min) FROM films WHERE length_min > 120;
  - GROUP BY clause
    - Ex: SELECT customer_id, count(*) FROM bookings
    - GROUP BY customer_id;
    - *To observe number of bookings for each customer_id*
  - HAVING clause
    - Ex: SELECT customer_id, screening_id, count(*) FROM bookings
    - GROUP BY customer_id, screening_id
    - HAVING customer_id = 10;
    - *In place of* WHERE *when using* GROUP BY
- MySQL Functions
  - Overview
    - *Functions can be passed parameters and return a value*
  - Concatenation
    - Ex: SELECT concat(first_name, ' ', last_name) AS full_name FROM customers;
  - Substrings
    - Ex: SELECT substring(name, 1, 4) AS short_name FROM films;
    - *substring(string, start, length)*
  - Case (Lower/Upper)
    - Ex: SELECT upper(name) AS name FROM rooms
  - Date
    - Ex: SELECT date(start_time) FROM screenings;
  - Month and Year
    - Ex: SELECT * FROM Screenings
    - WHERE month(start_time) = 4

- Subqueries
    - Overview
        - *Can be used in* SELECT*,* INSERT*,* UPDATE*, DELETE queries*
        - *Nested query can be in WHERE or FROM clause*
            - Non-Correlated Subqueries - *Inner query can run independently of outer query, and runs once*
        - Correlated Subqueries
            - *Inner query runs for every row returned by outer query*
    - Non-Correlated Subquery Example with WHERE
        - SELECT id, start_time FROM screenings
        - WHERE film_id IN
        - (SELECT id FROM films
        - WHERE length_min > 120)
        - ORDER BY start_time;
    - Non-Correlated Subquery Example with FROM
        - SELECT max(num_seats) FROM
        - (SELECT booking_id, count(seat_id) AS num_seats
        - FROM reserved_seat
        - GROUP BY booking_id) b;
        - *Note that the alias* b *is required for a derived table*
    - Correlated Subquery Example
        - SELECT screening_id, customer_id,
        - (SELECT count(*)
        - FROM reserved_seat WHERE booking_id = b.id)
        - FROM bookings b
        - ORDER BY screening_id;
        - *Can be more efficient than a correspondinq query utilizing* JOIN