|  |
| --- |
|  |
|  | Install shakila db from mysql site  ## 1. SELECT statements |
|  | # 1a. Select all columns from the actor table. |
|  | SELECT \* from actor; |
|  |  |
|  | # 1b. Select only the last\_name column from the actor table. |
|  | SELECT last\_name from actor; |
|  |  |
|  | # 1c. Select only the following columns from the film table. |
|  | # |
|  | # COLUMN NAME Note |
|  | # title Exists in film table. |
|  | # description Exists in film table. |
|  | # rental\_duration Exists in film table. |
|  | # rental\_rate Exists in film table. |
|  | # total\_rental\_cost rental\_duration \* rental\_rate |
|  |  |
|  | SELECT |
|  | title, |
|  | description, |
|  | rental\_duration, |
|  | rental\_rate, |
|  | rental\_duration \* rental\_rate as total\_rental\_cost |
|  | FROM film; |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 2. DISTINCT operator |
|  |  |
|  | # 2a. Select all distinct (different) last names from the actor table. |
|  | SELECT DISTINCT last\_name FROM actor; |
|  |  |
|  | # 2b. Select all distinct (different) postal codes from the address table. |
|  | SELECT DISTINCT postal\_code FROM address; |
|  |  |
|  |  |
|  | # 2c. Select all distinct (different) ratings from the film table. |
|  | SELECT DISTINCT rating FROM film; |
|  |  |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 3. WHERE clause |
|  |  |
|  | # 3a. Select the title, description, rating, movie length columns from the films table that last 3 hours or longer. |
|  | SELECT |
|  | title, |
|  | description, |
|  | rating, |
|  | length |
|  | FROM film |
|  | WHERE length >= 180; |
|  |  |
|  |  |
|  | # 3b. Select the payment id, amount, and payment date columns from the payments table for payments made on or after 05/27/2005. |
|  | SELECT |
|  | payment\_id, |
|  | amount, |
|  | payment\_date |
|  | FROM payment |
|  | WHERE payment\_date >= '2005-05-27'; |
|  |  |
|  |  |
|  | # 3c. Select the primary key, amount, and payment date columns from the payment table for payments made on 05/27/2005. |
|  | SELECT |
|  | payment\_id, |
|  | amount, |
|  | payment\_date |
|  | FROM payment |
|  | WHERE payment\_date >= '2005-05-27' AND payment\_date <= '2005-05-28'; |
|  |  |
|  |  |
|  | # 3d. Select all columns from the customer table for rows that have a last names beginning with S and a first names ending with N. |
|  | SELECT \* |
|  | FROM customer |
|  | WHERE last\_name LIKE 'S%' AND first\_name LIKE '%N'; |
|  |  |
|  |  |
|  | # 3e. Select all columns from the customer table for rows where the customer is inactive or has a last name beginning with "M". |
|  | SELECT \* |
|  | FROM customer |
|  | WHERE last\_name LIKE 'M%' OR active = "false"; |
|  |  |
|  |  |
|  | # 3f. Select all columns from the category table for rows where the primary key is greater than 4 and the name field begins with either C, S or T. |
|  | SELECT \* |
|  | FROM category |
|  | WHERE category\_id > 4 AND name LIKE 'C%' OR name LIKE 'S%' OR name LIKE 'T%'; |
|  |  |
|  | # 3g. Select all columns minus the password column from the staff table for rows that contain a password. |
|  | SELECT |
|  | staff\_id, |
|  | first\_name, |
|  | last\_name, |
|  | address\_id, |
|  | email, |
|  | picture, |
|  | store\_id, |
|  | active, |
|  | username, |
|  | last\_update |
|  | FROM staff |
|  | WHERE password IS NULL; |
|  |  |
|  |  |
|  | # 3h. Select all columns minus the password column from the staff table for rows that do not contain a password. |
|  | SELECT |
|  | staff\_id, |
|  | first\_name, |
|  | last\_name, |
|  | address\_id, |
|  | email, |
|  | picture, |
|  | store\_id, |
|  | active, |
|  | username, |
|  | last\_update |
|  | FROM staff |
|  | WHERE password IS NOT NULL; |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 4. IN operator |
|  |  |
|  | # 4a. Select the phone and district columns from the address table for addresses in California, England, Taipei, or West Java. |
|  | SELECT |
|  | phone, |
|  | district |
|  | FROM |
|  | address |
|  | WHERE district IN ('California', 'England', 'Taipei', 'West Java'); |
|  |  |
|  | # 4b. Select the payment id, amount, and payment date columns from the payment table for payments made on 05/25/2005, 05/27/2005, and 05/29/2005. |
|  | # (Use the IN operator and the DATE function, instead of the AND operator as in previous exercises.) |
|  | SELECT |
|  | payment\_id, |
|  | amount, |
|  | payment\_date |
|  | FROM |
|  | payment |
|  | WHERE |
|  | DATE(payment\_date) IN('2005-05-25', '2005-05-27', '2005-05-29'); |
|  |  |
|  | # 4c. Select all columns from the film table for films rated G, PG-13 or NC-17. |
|  | SELECT \* |
|  | FROM film |
|  | WHERE rating IN ('G', 'PG-13', 'NC-17'); |
|  |  |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 5. BETWEEN operator |
|  |  |
|  | # 5a. Select all columns from the payment table for payments made between midnight 05/25/2005 and 1 second before midnight 05/26/2005. |
|  | SELECT \* |
|  | FROM payment |
|  | WHERE payment\_date BETWEEN '2005-05-25' AND '2005-05-26'; |
|  |  |
|  |  |
|  | # 5b. Select the following columns from the film table for films where the length of the description is between 100 and 120. |
|  | # |
|  | # COLUMN NAME Note |
|  | # title Exists in film table. |
|  | # description Exists in film table. |
|  | # release\_year Exists in film table. |
|  | # total\_rental\_cost rental\_duration \* rental\_rate |
|  |  |
|  | SELECT |
|  | title, |
|  | description, |
|  | release\_year, |
|  | rental\_duration \* rental\_rate AS total\_rental\_cost |
|  | FROM film |
|  | WHERE length BETWEEN 100 AND 120; |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 6. LIKE operator |
|  |  |
|  | # 6a. Select the following columns from the film table for rows where the description begins with "A Thoughtful". |
|  | # Title, Description, Release Year |
|  | SELECT |
|  | title, |
|  | description, |
|  | release\_year |
|  | FROM film |
|  | WHERE description LIKE 'A Thoughtful%'; |
|  |  |
|  |  |
|  | # 6b. Select the following columns from the film table for rows where the description ends with the word "Boat". |
|  | # Title, Description, Rental Duration |
|  | SELECT |
|  | title, description, rental\_duration |
|  | FROM film |
|  | WHERE description LIKE '%Boat'; |
|  |  |
|  |  |
|  | # 6c. Select the following columns from the film table where the description contains the word "Database" and the length of the film is greater than 3 hours. |
|  | # Title, Length, Description, Rental Rate |
|  | SELECT |
|  | title, length, description, rental\_rate |
|  | FROM film |
|  | WHERE description LIKE '%Database%' && length > 180; |
|  |  |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 7. LIMIT Operator |
|  |  |
|  | # 7a. Select all columns from the payment table and only include the first 20 rows. |
|  | SELECT \* |
|  | FROM payment |
|  | LIMIT 20; |
|  |  |
|  | # 7b. Select the payment date and amount columns from the payment table for rows where the payment amount is greater than 5, and only select rows whose zero-based index in the result set is between 1000-2000. |
|  | SELECT |
|  | payment\_date, |
|  | payment\_id, |
|  | amount |
|  | FROM payment |
|  | WHERE amount > 5 |
|  | LIMIT 1000 offset 2000; |
|  |  |
|  | # 7c. Select all columns from the customer table, limiting results to those where the zero-based index is between 101-200. |
|  | SELECT \* |
|  | FROM customer |
|  | LIMIT 101 offset 200; |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 8. ORDER BY statement |
|  |  |
|  | # 8a. Select all columns from the film table and order rows by the length field in ascending order. |
|  | SELECT \* |
|  | FROM film |
|  | ORDER BY length ASC; |
|  |  |
|  |  |
|  | # 8b. Select all distinct ratings from the film table ordered by rating in descending order. |
|  | SELECT DISTINCT rating |
|  | FROM film |
|  | ORDER BY rating DESC; |
|  |  |
|  |  |
|  | # 8c. Select the payment date and amount columns from the payment table for the first 20 payments ordered by payment amount in descending order. |
|  | SELECT |
|  | payment\_date, |
|  | amount |
|  | FROM payment |
|  | ORDER BY amount DESC |
|  | LIMIT 20; |
|  |  |
|  |  |
|  | # 8d. Select the title, description, special features, length, and rental duration columns from the film table for the first 10 films with behind the scenes footage under 2 hours in length and a rental duration between 5 and 7 days, ordered by length in descending order. |
|  | SELECT |
|  | title, |
|  | description, |
|  | special\_features, |
|  | length, |
|  | rental\_duration |
|  | FROM film |
|  | WHERE rental\_duration BETWEEN 5 AND 7 AND length < 120 |
|  | ORDER BY length DESC |
|  | LIMIT 10; |
|  |  |
|  | # ---------------------------------------------------------# |
|  |  |
|  |  |
|  | ## 9. JOINS |
|  |  |
|  | # 9a. Select customer first\_name/last\_name and actor first\_name/last\_name columns from performing a /left join/ |
|  | # between the customer and actor column on the last\_name column in each table. |
|  | # (i.e. `customer.last\_name = actor.last\_name`) |
|  | # Label customer first\_name/last\_name columns as customer\_first\_name/customer\_last\_name |
|  | # Label actor first\_name/last\_name columns in a similar fashion. |
|  |  |
|  | SELECT |
|  | customer.first\_name AS customer\_first\_name, |
|  | customer.last\_name AS customer\_last\_name, |
|  | actor.first\_name AS actor\_first\_name, |
|  | actor.last\_name AS actor\_last\_name |
|  | FROM customer |
|  | LEFT JOIN actor ON customer.first\_name = actor.first\_name AND customer.last\_name = actor.last\_name; |
|  | # returns correct number of records: 599 |
|  |  |
|  | # 9b. Select the customer first\_name/last\_name and actor first\_name/last\_name columns from performing a /right join between the customer and actor column on the last\_name column in each table. (i.e. `customer.last\_name = actor.last\_name`) |
|  | SELECT |
|  | customer.first\_name AS customer\_first\_name, |
|  | customer.last\_name AS customer\_last\_name, |
|  | actor.first\_name AS actor\_first\_name, |
|  | actor.last\_name AS actor\_last\_name |
|  | FROM customer |
|  | RIGHT JOIN actor ON customer.first\_name = actor.first\_name AND customer.last\_name = actor.last\_name; |
|  | # returns correct number of records: 200 |
|  |  |
|  |  |
|  | # 9c. Select the customer first\_name/last\_name and actor first\_name/last\_name columns from performing an inner join between the customer and actor column on the last\_name column in each table. (i.e. `customer.last\_name = actor.last\_name`) |
|  | SELECT |
|  | customer.first\_name AS customer\_first\_name, |
|  | customer.last\_name AS customer\_last\_name, |
|  | actor.first\_name AS actor\_first\_name, |
|  | actor.last\_name AS actor\_last\_name |
|  | FROM customer |
|  | INNER JOIN actor ON customer.last\_name = actor.last\_name; |
|  | # returns correct number of records: 43 |
|  |  |
|  | # 9d. Select the city name and country name columns from the city table, performing a left join with the country table to get the country name column. |
|  | SELECT |
|  | city.city AS city, |
|  | country.country\_id AS country |
|  | FROM city |
|  | LEFT JOIN country ON city.country\_id = country.country\_id; |
|  | # Returns correct records: 600 |
|  |  |
|  |  |
|  | # 9e. Select the title, description, release year, and language name columns from the film table, performing a left join with the language table to get the "language" column. |
|  | # Label the language.name column as "language" (e.g. `select language.name as language`) |
|  | SELECT |
|  | film.title AS title, |
|  | film.description AS description, |
|  | film.release\_year, |
|  | film.language\_id AS language |
|  | FROM film |
|  | LEFT JOIN language ON language.language\_id = film.language\_id; |
|  | # returns 1000 rows: correct |
|  |  |
|  | # 9f. Select the first\_name, last\_name, address, address2, city name, district, and postal code columns from the staff table, performing 2 left joins with the address table then the city table to get the address and city related columns. |
|  | SELECT |
|  | staff.first\_name, |
|  | staff.last\_name, |
|  | staff.address\_id AS address, |
|  | address.address2, |
|  | address.city\_id AS city, |
|  | address.district, |
|  | address.postal\_code |
|  | FROM staff |
|  | LEFT JOIN address ON staff.address\_id = address.address\_id |
|  | LEFT JOIN city ON address.city\_id = city.city\_id; |
|  | # returns correct number of rows: 2 |