USE sakila

/\* See columns in actor \*/

SELECT \* FROM actor;

/\* QUERY FIRST AND LAST NAMES FROM ACTOR \*/

SELECT

first\_name

,last\_name

FROM

actor;

/\* Display the first and last name of each actor in a single column in upper case letters. Name the column `Actor Name`. \*/

SET SQL\_SAFE\_UPDATES = 0;

ALTER TABLE actor

ADD COLUMN Actor\_Name VARCHAR(100) AFTER actor\_id;

UPDATE actor SET Actor\_Name = CONCAT(first\_name, ' ', last\_name);

SELECT \* FROM actor;

/\*Find all actors whose last name contain the letters `GEN`: \*/

/\*SELECT last\_name FROM actor WHERE last\_name CONTAINS 'GEN';\*/

SELECT last\_name FROM actor WHERE last\_name LIKE "%GEN%";

/\* actors whose last names contain `LI`. Order the rows by last name and first name \*/

SELECT last\_name, first\_name FROM actor WHERE last\_name LIKE '%LI%'

/\*Using `IN`, display the `country\_id` and `country` columns of the following countries: Afghanistan, Bangladesh, and China:\*/

SELECT \* FROM country;

SELECT country\_id, country

FROM country

WHERE country IN ('Afghanistan', 'Bangladesh', 'China');

/\*create a column in the table `actor` named `description` and use the data type `BLOB`\*/

ALTER TABLE actor

ADD COLUMN description BLOB AFTER actor\_id;

SELECT \* FROM actor;

/\*Delete the `description` column\*/

ALTER TABLE actor DROP description;

SELECT \* FROM actor;

/\*List the last names of actors, as well as how many actors have that last name.\*/

SELECT last\_name, COUNT(\*) AS 'Actor last name'

FROM actor

GROUP BY last\_name;

/\* List the names of actors, only if they more than one\*/

SELECT last\_name, COUNT(\*) AS 'Actor last name'

FROM actor

GROUP BY last\_name

HAVING COUNT(\*) > 1

/\* Update "Groucho Williams with 'Harpo Williams'\*/

SELECT \* FROM actor Where Actor\_Name = 'Groucho Williams';

UPDATE actor

SET Actor\_Name = 'Harpo Williams', first\_name= 'Harpo'

WHERE actor\_id = 172;

/\* Update Harpo Williams to Groucho Williams and first names Groucho to Mucho Groucho\*/

SET SQL\_SAFE\_UPDATES = 0;

SELECT \* FROM actor Where first\_name = 'Mucho Groucho';

UPDATE actor SET first\_name = CASE

WHEN actor\_id = 172 THEN 'Groucho'

WHEN actor\_id = 106 THEN 'Mucho Groucho'

WHEN actor\_id = 78 THEN 'Mucho Groucho'

ELSE first\_name

END;

UPDATE actor SET Actor\_Name = CASE

WHEN Actor\_Name = 172 THEN 'Groucho Williams'

WHEN Actor\_Name = 106 THEN 'Mucho Groucho Dunst'

WHEN Actor\_Name = 78 THEN 'Mucho Groucho Sinatra'

ELSE Actor\_Name

END;

/\*You cannot locate the schema of the `address` table. Which query would you use to re-create it?\*/

Select \* FROM address;

CREATE TABLE `address` (

`address\_id` smallint(5) unsigned NOT NULL AUTO\_INCREMENT,

`address` varchar(50) NOT NULL,

`address2` varchar(50) DEFAULT NULL,

`district` varchar(20) NOT NULL,

`city\_id` smallint(5) unsigned NOT NULL,

`postal\_code` varchar(10) DEFAULT NULL,

`phone` varchar(20) NOT NULL,

`location` geometry NOT NULL,

`last\_update` timestamp NOT NULL DEFAULT CURRENT\_TIMESTAMP ON UPDATE CURRENT\_TIMESTAMP,

PRIMARY KEY (`address\_id`),

KEY `idx\_fk\_city\_id` (`city\_id`),

SPATIAL KEY `idx\_location` (`location`),

CONSTRAINT `fk\_address\_city` FOREIGN KEY (`city\_id`) REFERENCES `city` (`city\_id`) ON UPDATE CASCADE

) ENGINE=InnoDB AUTO\_INCREMENT=606 DEFAULT CHARSET=utf8

/\* Use `JOIN` to display the first and last names, as well as the address, of each staff member. Use the tables `staff` and `address`: \*/

select

first\_name

,last\_name

,address

from

staff

inner join address

on staff.address\_id = address.address\_id

/\*Using the tables `payment` and `customer` and the `JOIN` command, list the total paid by each customer.

List the customers alphabetically by last name:

from payment- customer\_id, amount

from customer- customer\_id, first\_name last\_name \*/

Select \*From payment;

Select \* From customer;

Select first\_name, last\_name, sum(payment.amount)

From customer

Inner Join payment on customer.customer\_id = payment.customer\_id

GROUP BY customer.customer\_id, last\_name;

/\*Use subqueries to display the titles of movies starting with the letters `K` and `Q` whose language is English. \*/

/\* Use film(title)\*/

Select \* From film;

Select title From film WHERE title LIKE 'Q\_%' or title Like 'K\_%'

/\*Use subqueries to display all actors who appear in the film `Alone Trip`\*/

Select \* from film\_actor; /\* film\_actor = actor id film\_id\*/

Select \* from film; /\* film= film\_id, title\*/

Select \* from actor; /\* = actor\_id, first\_name, last\_name\*/

SELECT first\_name, last\_name

FROM actor

WHERE actor\_id IN

(

SELECT actor\_id

FROM film\_actor

WHERE film\_id IN

(

SELECT film\_id

FROM film

WHERE title LIKE 'ALONE TRIP%'

)

);

/\* Identify all movies categorized as family films.\*/

Select \* from film; /\* film = film\_id, title\*/

Select \* from category; /\*category = category\_id, name(Family)\*/

Select \* from film\_category; /\*film\_id, category\_id\*/

SELECT title

FROM film

WHERE film\_id IN

(

SELECT film\_id

FROM film\_category

WHERE category\_id IN

(

SELECT category\_id

FROM category

WHERE name LIKE 'Family%'

)

);