**SQLBOLT**

|  |
| --- |
| EXERCISE 1:    SELECT title FROM movies;  SELECT director FROM movies;  SELECT title, director FROM movies;  SELECT title, year FROM movies;  SELECT \* FROM movies; |
| EXERCISE 2:    SELECT \* FROM movies WHERE id=6;  SELECT \* FROM movies WHERE year BETWEEN 2000 AND 2010;  SELECT \* FROM movies WHERE year NOT BETWEEN 2000 AND 2010;  SELECT title, year FROM movies WHERE id BETWEEN 1 AND 5; |

|  |
| --- |
| EXERCISE 3:    SELECT \* FROM movies WHERE title LIKE "%toy%";  SELECT \* FROM movies WHERE director LIKE "%john%";  SELECT \* FROM movies WHERE director not LIKE "%john%";  SELECT \* FROM movies WHERE title LIKE "%wall%"; |
| EXERCISE 4:    SELECT DISTINCT director FROM movies ORDER BY director ASC;  SELECT \* FROM movies ORDER BY year DESC LIMIT 4;  SELECT \* FROM movies ORDER BY title ASC LIMIT 5;  SELECT \* FROM movies ORDER BY title ASC LIMIT 5 OFFSET 5; |

|  |
| --- |
| EXERCISE 5:    SELECT \* FROM north\_american\_cities WHERE country LIKE "%canada%";  SELECT \* FROM north\_american\_cities WHERE country LIKE "%united states%" ORDER BY latitude DESC;  SELECT \*FROM North\_american\_cities WHERE longitude < (SELECT longitude FROM North\_american\_cities WHERE city = 'Chicago') ORDER BY longitude ASC;  SELECT \*FROM North\_american\_cities WHERE country LIKE "%Mexico%" ORDER BY population DESC LIMIT 2;  SELECT \*FROM North\_american\_cities WHERE country LIKE "%UNITED%" ORDER BY population DESC LIMIT 2 OFFSET 2; |
| EXERCISE 6:    SELECT \* FROM movies INNER JOIN boxoffice on movies.id = boxoffice.movie\_id;  SELECT \* FROM movies INNER JOIN boxoffice on movies.id = boxoffice.movie\_id WHERE international\_sales > domestic\_sales;  SELECT \* FROM movies INNER JOIN boxoffice on movies.id = boxoffice.movie\_id ORDER BY RATING DESC; |

|  |
| --- |
| EXERCISE 7:    SELECT \* FROM buildings JOIN employees ON buildings.building\_name = employees.building GROUP BY buildings.building\_name ;  SELECT \* FROM buildings;  SELECT \* FROM buildings LEFT JOIN employees ON buildings.building\_name = employees.building GROUP BY buildings.building\_name, employees.role  ORDER BY buildings.building\_name; |
| EXERCISE 8:    SELECT \* FROM employees WHERE building IS NULL;  SELECT building\_name FROM buildings LEFT JOIN employees ON buildings.building\_name = employees.building WHERE employees.building IS NULL; |

|  |
| --- |
| EXERCISE 9:    SELECT \*, SUM(CAST(boxoffice.domestic\_sales+boxoffice.international\_sales AS FLOAT))/1000000 AS millions FROM movies LEFT JOIN boxoffice ON movies.id = boxoffice.movie\_id GROUP BY movies.title;  SELECT \*, boxoffice.rating\*10 AS percent FROM movies LEFT JOIN boxoffice ON movies.id = boxoffice.movie\_id GROUP BY movies.title;  SELECT \* FROM MOVIES WHERE year % 2 = 0; |
| EXERCISE 10:    SELECT MAX(years\_employed) FROM employees ;  SELECT \*, AVG(years\_employed) FROM employees GROUP BY ROLE ;  SELECT \*, SUM(YEARS\_EMPLOYED) FROM employees GROUP BY building; |

|  |
| --- |
| EXERCISE 11:    SELECT \*, COUNT(\*) FROM employees WHERE role = 'Artist';  SELECT \*, COUNT(\*) FROM employees GROUP BY role ;  SELECT \*, SUM(years\_employed) FROM employees WHERE role = 'Engineer' ; |
| EXERCISE 12:    SELECT \*, COUNT(\*) FROM movies GROUP BY director;  SELECT \*, SUM(boxoffice.domestic\_sales+boxoffice.international\_sales) AS sales FROM movies JOIN boxoffice ON movies.id = boxoffice.movie\_id GROUP BY director; |

|  |
| --- |
| EXERCISE 13:    INSERT INTO movies (id, title, director, year, length\_minutes)  VALUES (4,'Toy Story 4', 'John Lasseter', 2019, 110);  INSERT INTO BoxOffice (movie\_id, rating, domestic\_sales, international\_sales)  VALUES ((SELECT id FROM movies WHERE title = 'Toy Story 4'), 8.7, 340000000, 270000000); |
| EXERCISE 14:    UPDATE movies SET director = 'John Lasseter' WHERE title = 'A Bug''s Life';  UPDATE movies SET year = 1999 WHERE title = 'Toy Story 2';  UPDATE movies SET title = 'Toy Story 3', director = 'Lee Unkrich' WHERE title = 'Toy Story 8'; |

|  |
| --- |
| EXERCISE 15:    DELETE FROM Movies WHERE year < 2005;  DELETE FROM Movies WHERE director = 'Andrew Stanton'; |
| EXERCISE 16:    CREATE TABLE Database (  Name TEXT,  Version FLOAT,  Download\_count INTEGER); |

|  |
| --- |
| EXERCISE 17:    ALTER TABLE movies  ADD COLUMN Aspect\_ratio FLOAT;  ALTER TABLE movies  ADD COLUMN Language TEXT DEFAULT 'English'; |
| EXERCISE 18:    DROP TABLE Movies;  DROP TABLE boxoffice; |