**SQL Lesson 1: SELECT queries 101**

A screenshot of a computer

Description automatically generated

Answers:

Q1. SELECT title FROM movies;

Q2. SELECT director FROM movies;

Q3. SELECT title , director FROM movies;

Q4. SELECT title , year FROM movies;

Q5. SELECT \* FROM movies;

**SQL Lesson 2: Queries with constraints (Pt. 1)**

**A screenshot of a computer

Description automatically generated**

ANSWERS:

Q1. SELECT \* FROM movies WHERE rowid = 6;

Q2. SELECT \* FROM movies WHERE year between 2000 and 2010;

Q3. SELECT \* FROM movies WHERE year not between 2000 and 2010;

Q4. SELECT title FROM movies WHERE id in (1,2,3,4,5) ;

**SQL Lesson 3: Queries with constraints (Pt. 2)**

**A screenshot of a computer

Description automatically generated**

ANSWERS:

Q1. SELECT \* FROM movies WHERE title like "TOY STORY%";

Q2. SELECT \* FROM movies WHERE director like "John Lasseter";

Q3. SELECT \* FROM movies WHERE director not like "John Lasseter";

Q4. SELECT \* FROM movies WHERE title like "WALL%";

**SQL Lesson 4: Filtering and sorting Query result**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT distinct director FROM movies order by director asc ;

Q2. SELECT title FROM movies order by year desc limit 4;

Q3. SELECT title FROM movies order by title asc limit 5;

Q4. SELECT title FROM movies order by title asc limit 5 offset 5;

**SQL Review: Simple SELECT Queries**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT city, population FROM north\_american\_cities WHERE country like "canada";

Q2. SELECT city, population FROM north\_american\_cities WHERE country like "United States" order by latitude desc;

Q3. SELECT city FROM north\_american\_cities WHERE longitude<-87.629798 order by longitude asc;

Q4. SELECT city,population FROM north\_american\_cities WHERE country like “Mexico” order by population desc limit 2;

Q5. SELECT city, population FROM north\_american\_cities WHERE country like "UNITED STATES" ORDER BY population DESC LIMIT 2 OFFSET 2;

**SQL Lesson 6: Multi-table queries with JOIN**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT title, domestic\_sales, international\_sales FROM movies JOIN boxoffice ON movies.id=boxoffice.movie\_id;

Q2. SELECT title, domestic\_sales, international\_sales FROM movies JOIN boxoffice ON movies.id=boxoffice.movie\_id WHERE international\_sales>domestic\_sales;

Q3. SELECT title, rating FROM movies JOIN boxoffice ON movies.id=boxoffice.movie\_id order by rating desc;

**SQL Lesson 7: OUTER JOINs**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT distinct building FROM employees;

Q2. SELECT \* FROM buildings;

Q3. SELECT distinct building\_name, role FROM buildings Left join employees On building\_name=building;

**SQL Lesson 8: A short note on NULLs**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT name, role FROM employees WHERE building is null;

Q2. SELECT distinct building\_name FROM buildings Left join employees ON building\_name= building WHERE role is null;

**SQL Lesson 9: Queries with expressions**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT title, (domestic\_sales + international\_sales) / 1000000 AS gross\_sales\_millions FROM movies join boxoffice on movies.id = boxoffice.movie\_id;

Q2. SELECT title, rating \* 10 AS rating\_percent FROM movies join boxoffice on movies.id = boxoffice.movie\_id;

Q3. SELECT title,year FROM movies WHERE year%2=0;

**SQL Lesson 10: Queries with aggregates (Pt. 1)** A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT max(years\_employed) as max\_years\_employed FROM employees;

Q2. SELECT role, AVG(years\_employed) as Average\_years\_employed FROM employees Group by role;

Q3. SELECT building, SUM(years\_employed) as Total\_years\_employed FROM employees Group by building;

**SQL Lesson 11: Queries with aggregates (Pt. 2)**

**A screenshot of a computer

Description automatically generated**

ANSWERS:

Q1. SELECT role, count(\*) as Number\_of\_artists FROM employees Where role = "Artist";

Q2. SELECT role,count(\*)FROM employees Group by role;

Q3. SELECT role,SUM(years\_employed) FROM employees Group by role Having role = "Engineer";

**SQL Lesson 12: Order of execution of a Query** A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. SELECT director,count (id) as Num\_movies\_directed FROM movies Group by director;

Q2. SELECT director,sum(domestic\_sales + international\_sales) as

Cumulative\_sales\_from\_all\_movies FROM movies Inner join boxoffice On

movies.id = boxoffice.movie\_id Group by director;

**SQL Lesson 13: Inserting rows**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. INSERT INTO movies values (4,"Toy Story 4","John Lasseter",2004,95);

Q2. INSERT INTO boxoffice values(4,8.7,340000000, 270000000);

**SQL Lesson 14: Updating rows**

A screenshot of a computer

Description automatically generated

ANSWERS:

Q1. UPDATE movies SET director = "John Lasseter" WHERE id=2;

Q2. UPDATE movies SET year =1999 WHERE title="Toy Story 2";

Q3. UPDATE movies SET title ="Toy Story 3",director =" Lee Unkrich" WHERE title ="Toy Story 8";

**SQL Lesson 15: Deleting rows**

**A screenshot of a computer

Description automatically generated**

ANSWERS:

Q1. DELETE FROM movies WHERE year<2005;

Q2. DELETE FROM movies WHERE director="Andrew Stanton";

**SQL Lesson 16: Creating tables**

**A screenshot of a computer program

Description automatically generated**

ANSWERS:

Q1. CREATE TABLE DATABASE (

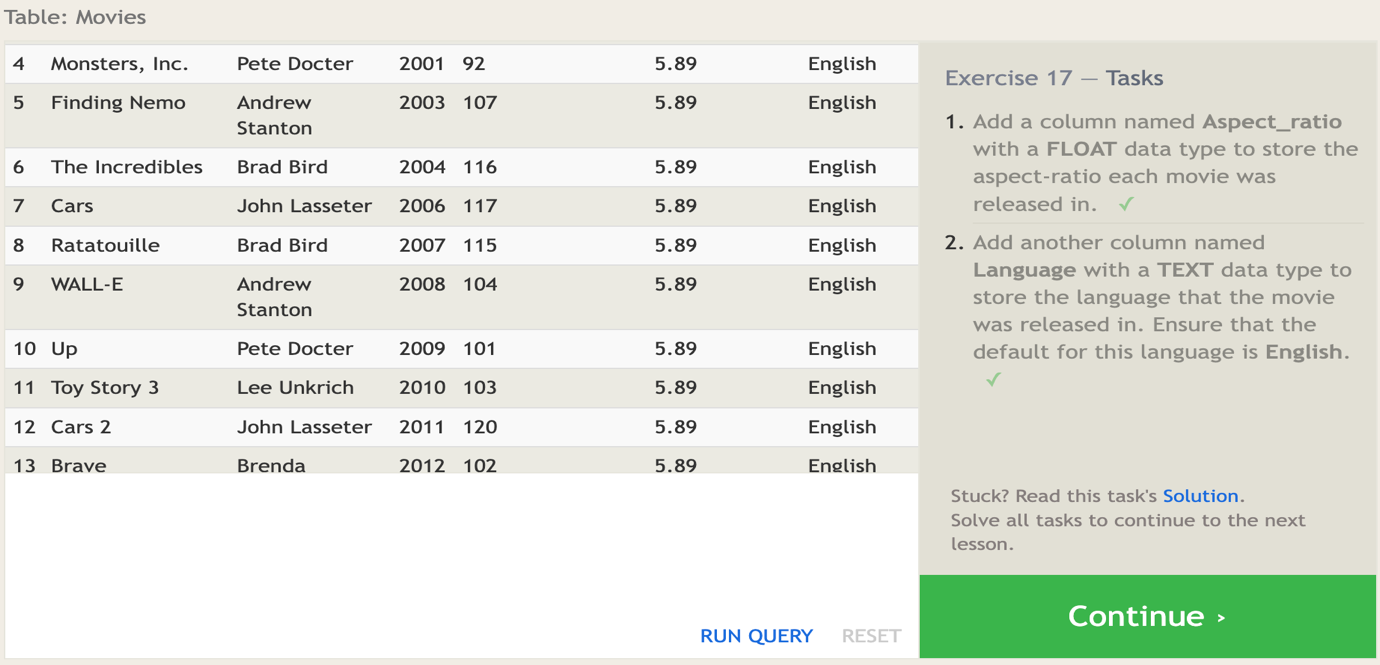
id INTEGER PRIMARY KEY,

NAME TEXT,

VERSION FLOAT,

Download\_count INTEGER );

**SQL Lesson 17: Altering tables**

****

ANSWERS:

Q1. ALTER TABLE movies ADD column Aspect\_ratio float DEFAULT 5.89;

Q2. ALTER TABLE movies ADD column Language TEXT DEFAULT "English";

**SQL Lesson 18: Dropping tables**

**A screenshot of a computer

Description automatically generated**

ANSWERS:

Q1. DROP TABLE IF EXISTS movies;

Q2. DROP TABLE IF EXISTS boxoffice;

A black graduation cap with a lightning bolt

Description automatically generated