SELECT basics tutorials:

1. SELECT population FROM world WHERE name = 'Germany';
2. SELECT name, population FROM world WHERE name IN ('Sweden', 'Norway', 'Denmark');
3. SELECT name, area FROM world WHERE area BETWEEN 200000 AND 250000

SELECT from world tutorials:

1. SELECT name, continent, population FROM world
2. SELECT name FROM world WHERE population > 200000000
3. SELECT name, GDP/population FROM world WHERE population > 200000000
4. SELECT name, population/1000000 FROM world WHERE continent = 'South America'
5. SELECT name, population FROM world WHERE name IN ('France', 'Germany', 'Italy')
6. SELECT name FROM world WHERE name LIKE '%United%'
7. SELECT name, population, area FROM world WHERE area > 3000000 OR population > 250000000
8. SELECT name, population, area FROM world WHERE (area > 3000000 AND population < 250000000) OR (area < 3000000 AND population > 250000000)
9. SELECT name, population, gdp FROM world WHERE continent = 'South America'
10. SELECT name, ROUND(gdp/population, -3) FROM world WHERE gdp > 1000000000000
11. SELECT name, LEN(name), continent, LEN(continent), capital, LEN(capital) FROM world WHERE name LEN(name) = LEN(capital)
12. SELECT name, capital FROM world WHERE LEFT(name,1) = LEFT(capital,1) AND name <> capital
13. SELECT name

FROM world

WHERE name LIKE '%A%'

AND name LIKE '%E%'

AND name LIKE '%I%'

AND name LIKE '%O%'

AND name LIKE '%U%'

AND name LIKE '%a%'

AND name LIKE '%e%'

AND name LIKE '%i%'

AND name LIKE '%o%'

AND name LIKE '%u%'

AND name NOT LIKE '% %'

SUM and COUNT tutorial:

1. SELECT SUM(population) FROM world
2. SELECT distinct continent FROM world
3. SELECT SUM(gdp) FROM world WHERE continent = 'Africa'
4. SELECT COUNT(name) FROM world WHERE area > 1000000
5. SELECT SUM(population) FROM world WHERE name IN ('Estonia', 'Latvia', 'Lithuania')
6. SELECT continent, COUNT(name) FROM world GROUP BY continent
7. SELECT continent, COUNT(name) FROM world WHERE population >= 10000000 GROUP BY continent

Nobel Tutorial:

1. SELECT yr, subject, winner FROM nobel WHERE yr = 1950;
2. SELECT winner FROM nobel WHERE yr = 1962 AND subject = 'literature'
3. SELECT yr, subject FROM nobel WHERE winner = 'Albert Einstein'
4. SELECT winner FROM nobel WHERE subject = 'peace' AND yr >= 2000
5. SELECT yr, subject, winner FROM nobel WHERE subject = 'literature' AND yr BETWEEN 1980 AND 1989;
6. SELECT \* FROM nobel WHERE winner IN ('Theodore Roosevelt', 'Woodrow Wilson', 'Jimmy Carter', 'Barack Obama')
7. SELECT winner FROM nobel WHERE winner LIKE 'John%'
8. SELECT \* FROM nobel WHERE (subject = 'Physics' AND yr = '1980') OR (subject = 'Chemistry' AND yr = 1984)
9. SELECT \* FROM nobel WHERE yr = 1980 AND subject NOT IN('Chemistry', 'Medicine')
10. SELECT \* FROM nobel WHERE (subject = 'Medicine' AND yr < '1910') OR (subject = 'Literature' AND yr >= '2004')
11. SELECT \* FROM nobel WHERE winner = 'PETER GRÜNBERG'
12. SELECT \* FROM nobel WHERE winner = 'EUGENE O''NEILL
13. SELECT winner, yr, subject FROM nobel WHERE winner LIKE 'Sir%' ORDER BY yr DESC, winner
14. SELECT winner, subject, subject IN ('Physics','Chemistry') FROM nobel WHERE yr=1984 ORDER BY subject IN ('Physics', 'Chemistry'), subject, winner

Parts 3: Products Querying

1. INSERT INTO products (name, price, can\_be\_returned) VALUES ('chair', 44.00, false);
2. INSERT INTO products (name, price, can\_be\_returned) VALUES ('stool', 25.99, true);
3. INSERT INTO products (name, price, can\_be\_returned) VALUES ('table', 124.00, false);
4. SELECT \* FROM products;
5. SELECT name FROM products;
6. SELECT name, price FROM products;
7. INSERT INTO products (name, price, can\_be\_returned) VALUES ('couch', 1400.00, true);
8. SELECT name FROM products WHERE can\_be\_returned;
9. SELECT name FROM products WHERE price < 44.00;
10. SELECT name FROM products WHERE price BETWEEN 22.50 AND 99.99;
11. UPDATE products SET price = price - 20;
12. DELETE FROM products WHERE price < 25;
13. UPDATE products SET price = price + 20;
14. UPDATE products SET can\_be\_returned = true;

Part 4: Google Play Store Querying

1. SELECT app\_name FROM analytics WHERE id = 1880;
2. SELECT id, app\_name FROM analytics WHERE last\_updated = 'August 01, 2018';
3. SELECT category, COUNT(\*) FROM analytics GROUP BY category;
4. SELECT \* FROM analytics ORDER BY reviews DESC LIMIT 5;
5. SELECT \* FROM analytics WHERE rating >= 4.8 ORDER BY reviews DESC LIMIT 1;
6. SELECT category, AVG(rating) FROM analytics GROUP BY category ORDER BY AVG(rating) DESC;
7. SELECT app\_name, price, rating FROM analytics WHERE rating < 3 ORDER BY price DESC LIMIT 1;
8. SELECT \* FROM analytics WHERE min\_installs <= 50 AND rating > 0 ORDER BY rating DESC;
9. SELECT app\_name FROM analytics WHERE rating < 3 AND reviews = 10000;
10. SELECT \* FROM analytics WHERE price BETWEEN 0.10 AND 1.00 ORDER BY reviews DESC LIMIT 10;
11. SELECT \* FROM analytics ORDER BY last\_updated LIMIT 10;
12. SELECT \* FROM analytics ORDER by price DESC LIMIT 1;
13. SELECT COUNT(reviews) FROM analytics;
14. SELECT category FROM analytics GROUP BY category HAVING COUNT(\*) > 300;
15. SELECT app\_name, reviews, min\_installs, min\_installs/reviews AS proportion FROM analytics WHERE min\_installs >=100000 ORDER BY proportion DESC LIMIT 1;