PROJECT TITLE: **Analyzing Heart Attack Risk Patients' Data**

Dataset Link: **https://www.kaggle.com/datasets/iamsouravbanerjee/heart-attack-prediction-dataset**

1. DATA ACCESS AND SQLLDR

Replicate here the SQLLDR task that you use to ingest the raw data from the proposed CSV file:

- **CREATE TABLE**

  SQL:

```
CREATE TABLE HEART_ATTACK_RISK_PATIENT_INFO  (
   Patient_ID VARCHAR2(128),
        Age NUMBER,
        Sex VARCHAR2(128),
   Cholesterol NUMBER,
   Blood_Pressure VARCHAR2(128),
   Heart_Rate NUMBER,
   Diabetes NUMBER,
   Family_History NUMBER,
   Smoking NUMBER,
        Obesity NUMBER,
        Alcohol_Consumption NUMBER,
        Exercise_Hours_Per_Week NUMBER,
        Diet VARCHAR2(128),
        Previous_Heart_Problems NUMBER,
        Medication_Use NUMBER,
        Stress_Level NUMBER,
        Sedentary_Hours_Per_Day NUMBER,
        Income NUMBER,
        BMI NUMBER,
        Triglycerides NUMBER,
        Physical_Activity_Days_Per_Week NUMBER,
        Sleep_Hours_Per_Day NUMBER,
        Country VARCHAR2(128),
        Continent VARCHAR2(128),
        Hemisphere VARCHAR2(128),
        Heart_Attack_Risk NUMBER
);
```

- **Control File**

  SQL:

```
OPTIONS (SKIP=1)
LOAD DATA
CHARACTERSET UTF8
```

```
INFILE "heart_attack_prediction_dataset.csv" "str '\r\n'"
APPEND INTO TABLE HEART_ATTACK_RISK_PATIENT_INFO
FIELDS TERMINATED BY ',' OPTIONALLY ENCLOSED BY ""
TRAILING NULLCOLS
(
   Patient_ID CHAR,
   Age INTEGER EXTERNAL,
   Sex CHAR,
   Cholesterol INTEGER EXTERNAL,
   Blood_Pressure CHAR,
   Heart_Rate INTEGER EXTERNAL,
   Diabetes INTEGER EXTERNAL,
   Family_History INTEGER EXTERNAL,
   Smoking INTEGER EXTERNAL,
   Obesity INTEGER EXTERNAL,
   Alcohol_Consumption INTEGER EXTERNAL,
   Exercise_Hours_Per_Week DECIMAL EXTERNAL,
   Diet CHAR,
   Previous_Heart_Problems INTEGER EXTERNAL,
   Medication_Use INTEGER EXTERNAL,
   Stress_Level INTEGER EXTERNAL,
   Sedentary_Hours_Per_Day DECIMAL EXTERNAL,
   Income INTEGER EXTERNAL,
   BMI DECIMAL EXTERNAL,
   Triglycerides INTEGER EXTERNAL,
   Physical_Activity_Days_Per_Week INTEGER EXTERNAL,
   Sleep_Hours_Per_Day INTEGER EXTERNAL,
   Country CHAR,
   Continent CHAR,
   Hemisphere CHAR,
   Heart_Attack_Risk INTEGER EXTERNAL
)
```

- **SQLLDR command-line**
  1) Upload the heart_attack_prediction_dataset.csv file to your Linux account (replace "abc123" with your username) – your Linux password will be required.

     scp heart_attack_prediction_dataset.csv abc123@linux.cci.drexel.edu:~

  2) Upload the heart_attack_risk_prediction.ctl file (same path as you have heart_attack_prediction_dataset.csv), so you will be able to use SQLLDR to ingest the data (replace "abc123" with your username) – your Linux password will be required.

     scp heart_attack_risk_prediction.ctl abc123@linux.cci.drexel.edu:~

  3) Connect to your Linux account:

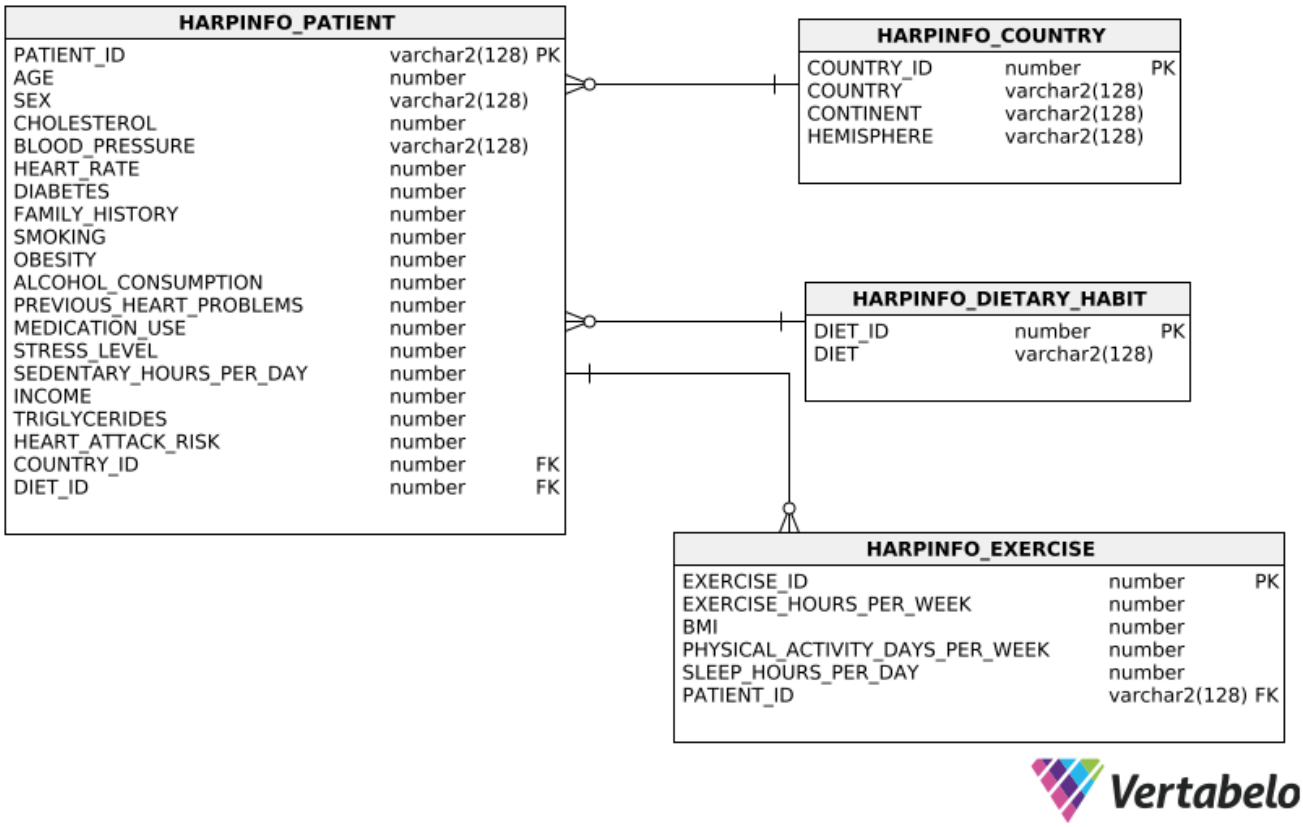
     ssh abc123@linux.cci.drexel.edu

4) Run the following command (replace "abc123" with your username):

sqlldr abc123@ike control=heart_attack_risk_prediction.ctl

2. DATA NORMALIZATION

Physical Data Model of Our Normalized Database:



3. DEPLOYMENT

Export DDL from Vertabelo.com and create your database. Add a full DDL script below, including any changes made to the original SQL DDL exported from Vertabelo.com.

- Start with DROP TABLE statements so your full script can be tested multiple times.

SQL DDL For HARPINFO_COUNTRY Table

```
DROP TABLE HARPINFO_COUNTRY CASCADE CONSTRAINTS;

CREATE TABLE HARPINFO_COUNTRY (
    COUNTRY_ID NUMBER PRIMARY KEY,
    COUNTRY VARCHAR2(128),
    CONTINENT VARCHAR2(128),
```

```
    HEMISPHERE VARCHAR2(128)
);
```

SQL DDL For HARPINFO_DIETARY_HABIT Table

```
DROP TABLE HARPINFO_DIETARY_HABIT CASCADE CONSTRAINTS;

CREATE TABLE HARPINFO_DIETARY_HABIT (
    DIET_ID NUMBER PRIMARY KEY,
    DIET VARCHAR2(128)
);
```

SQL DDL For HARPINFO_PATIENT Table

```
DROP TABLE HARPINFO_PATIENT CASCADE CONSTRAINTS;

CREATE TABLE HARPINFO_PATIENT (
    PATIENT_ID VARCHAR2(128) PRIMARY KEY,
    AGE NUMBER,
    SEX VARCHAR2(128),
    CHOLESTEROL NUMBER,
    BLOOD_PRESSURE VARCHAR2(128),
    HEART_RATE NUMBER,
    DIABETES NUMBER,
    FAMILY_HISTORY NUMBER,
    SMOKING NUMBER,
    OBESITY NUMBER,
    ALCOHOL_CONSUMPTION NUMBER,
    PREVIOUS_HEART_PROBLEMS NUMBER,
    MEDICATION_USE NUMBER,
    STRESS_LEVEL NUMBER,
    SEDENTARY_HOURS_PER_DAY NUMBER,
    INCOME NUMBER,
    TRIGLYCERIDES NUMBER,
    HEART_ATTACK_RISK NUMBER,
    COUNTRY_ID NUMBER,
    DIET_ID NUMBER,
    FOREIGN KEY (COUNTRY_ID) REFERENCES HARPINFO_COUNTRY(COUNTRY_ID),
    FOREIGN KEY (DIET_ID) REFERENCES HARPINFO_DIETARY_HABIT(DIET_ID)
);
```

SQL DDL For HARPINFO_EXERCISE Table

```
DROP TABLE HARPINFO_EXERCISE CASCADE CONSTRAINTS;

CREATE TABLE HARPINFO_EXERCISE (
    EXERCISE_ID NUMBER PRIMARY KEY,
    EXERCISE_HOURS_PER_WEEK NUMBER,
    BMI NUMBER,
```

```
    PHYSICAL_ACTIVITY_DAYS_PER_WEEK NUMBER,
    SLEEP_HOURS_PER_DAY NUMBER,
    PATIENT_ID VARCHAR2(128),
    FOREIGN KEY (PATIENT_ID) REFERENCES HARPINFO_PATIENT(PATIENT_ID)
);
```

4. DATA INGESTION

Write SQL DML statements to ingest data from the original tables into the proposed normalized database.

- You can use as many INSERT, DELETE, UPDATE to ingest data
- You can't CREATE any additional tables during this process
- You can use SELECT in between the process to check whether ingestion process is correct (include screenshots)

SQL DML For "HARPINFO_COUNTRY" Table

```
INSERT INTO HARPINFO_COUNTRY (COUNTRY_ID, COUNTRY, CONTINENT,
HEMISPHERE)
SELECT
    ROW_NUMBER() OVER (ORDER BY Country) AS COUNTRY_ID,
    Country,
    Continent,
    Hemisphere
FROM
    (SELECT DISTINCT Country, Continent, Hemisphere FROM
HEART_ATTACK_RISK_PATIENT_INFO);
```

TESTING:

```
SELECT * FROM HARPINFO_COUNTRY;
```

Output:

Script Output x | Query Result x

SQL | All Rows Fetched: 20 in 0.022 seconds

	COUNTRY_ID	COUNTRY	CONTINENT	HEMISPHERE
1	1	Argentina	South America	Southern Hemisphere
2	2	Australia	Australia	Southern Hemisphere
3	3	Brazil	South America	Southern Hemisphere
4	4	Canada	North America	Northern Hemisphere
5	5	China	Asia	Northern Hemisphere
6	6	Colombia	South America	Northern Hemisphere
7	7	France	Europe	Northern Hemisphere
8	8	Germany	Europe	Northern Hemisphere
9	9	India	Asia	Northern Hemisphere
10	10	Italy	Europe	Southern Hemisphere
11	11	Japan	Asia	Northern Hemisphere
12	12	New Zealand	Australia	Southern Hemisphere
13	13	Nigeria	Africa	Northern Hemisphere
14	14	South Africa	Africa	Southern Hemisphere
15	15	South Korea	Asia	Northern Hemisphere
16	16	Spain	Europe	Southern Hemisphere
17	17	Thailand	Asia	Northern Hemisphere
18	18	United Kingdom	Europe	Northern Hemisphere
19	19	United States	North America	Northern Hemisphere
20	20	Vietnam	Asia	Northern Hemisphere

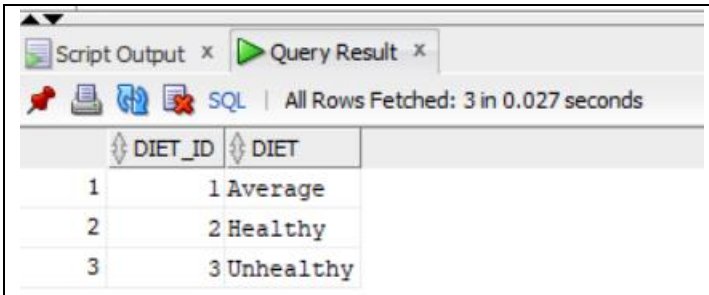
SQL DML For "HARPINFO_DIETARY_HABIT" Table

```
INSERT INTO HARPINFO_DIETARY_HABIT (DIET_ID, DIET)
SELECT
    ROW_NUMBER() OVER (ORDER BY Diet) AS DIET_ID,
    Diet
FROM
    (SELECT DISTINCT Diet FROM HEART_ATTACK_RISK_PATIENT_INFO);
```

TESTING:

```
SELECT * FROM HARPINFO_DIETARY_HABIT;
```

Output:



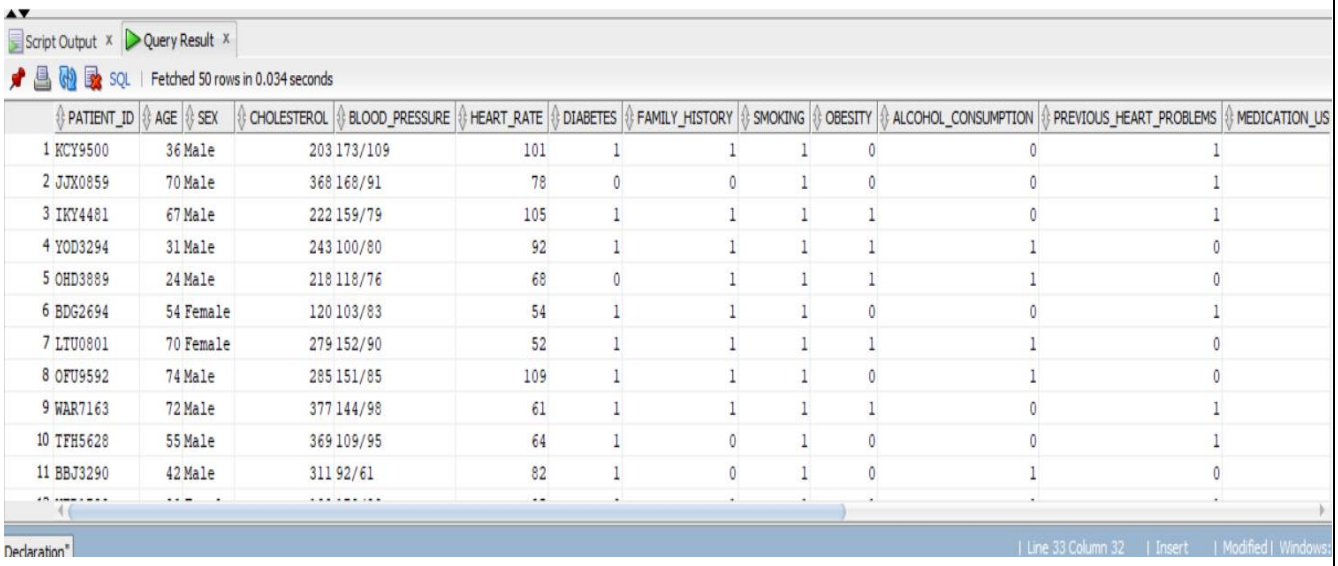
	DIET_ID	DIET
1	1	Average
2	2	Healthy
3	3	Unhealthy

SQL DML For "HARPINFO_PATIENT" Table

```
INSERT INTO HARPINFO_PATIENT (
    PATIENT_ID, AGE, SEX, CHOLESTEROL, BLOOD_PRESSURE, HEART_RATE, DIABETES,
FAMILY_HISTORY,
    SMOKING,    OBESITY,    ALCOHOL_CONSUMPTION,    PREVIOUS_HEART_PROBLEMS,
MEDICATION_USE,
    STRESS_LEVEL,    SEDENTARY_HOURS_PER_DAY,    INCOME,    TRIGLYCERIDES,
HEART_ATTACK_RISK,
    COUNTRY_ID, DIET_ID
)
SELECT
    Patient_ID,
    Age,
    Sex,
    Cholesterol,
    Blood_Pressure,
    Heart_Rate,
    Diabetes,
    Family_History,
    Smoking,
    Obesity,
    Alcohol_Consumption,
    Previous_Heart_Problems,
    Medication_Use,
    Stress_Level,
    Sedentary_Hours_Per_Day,
    Income,
    Triglycerides,
    Heart_Attack_Risk,
    (SELECT    COUNTRY_ID    FROM    HARPINFO_COUNTRY    WHERE    COUNTRY    =
HEART_ATTACK_RISK_PATIENT_INFO.Country),
    (SELECT    DIET_ID    FROM    HARPINFO_DIETARY_HABIT    WHERE    DIET    =
HEART_ATTACK_RISK_PATIENT_INFO.Diet)
FROM
    HEART_ATTACK_RISK_PATIENT_INFO;

TESTING:
SELECT * FROM HARPINFO_PATIENT;
```

Output:



Script Output | Query Result

SQL | Fetched 50 rows in 0.034 seconds

	PATIENT_ID	AGE	SEX	CHOLESTEROL	BLOOD_PRESSURE	HEART_RATE	DIABETES	FAMILY_HISTORY	SMOKING	OBESITY	ALCOHOL_CONSUMPTION	PREVIOUS_HEART_PROBLEMS	MEDICATION_US
1	KCY9500	36	Male	203	173/109	101	1	1	1	0	0	1	
2	JJX0859	70	Male	368	168/91	78	0	0	1	0	0	1	
3	IKY4481	67	Male	222	159/79	105	1	1	1	1	0	1	
4	YOD3294	31	Male	243	100/80	92	1	1	1	1	1	0	
5	OHD3889	24	Male	218	118/76	68	0	1	1	1	1	0	
6	BDG2694	54	Female	120	103/83	54	1	1	1	0	0	1	
7	LIU0801	70	Female	279	152/90	52	1	1	1	1	1	0	
8	OFU9592	74	Male	285	151/85	109	1	1	1	0	1	0	
9	WAR7163	72	Male	377	144/98	61	1	1	1	1	0	1	
10	TFH5628	55	Male	369	109/95	64	1	0	1	0	0	1	
11	BBJ3290	42	Male	311	92/61	82	1	0	1	0	1	0	

Declaration* | Line 33 Column 32 | Insert | Modified | Windows

SQL DML For "HARPINFO_EXERCISE" Table

```
INSERT INTO HARPINFO_EXERCISE (
    EXERCISE_ID,                EXERCISE_HOURS_PER_WEEK,                BMI,
PHYSICAL_ACTIVITY_DAYS_PER_WEEK,
    SLEEP_HOURS_PER_DAY, PATIENT_ID
)
SELECT
    ROW_NUMBER() OVER (ORDER BY Patient_ID) AS EXERCISE_ID,
    Exercise_Hours_Per_Week,
    BMI,
    Physical_Activity_Days_Per_Week,
    Sleep_Hours_Per_Day,
    Patient_ID
FROM
    HEART_ATTACK_RISK_PATIENT_INFO;
```

TESTING:

```
SELECT * FROM HARPINFO_EXERCISE;
```

Output:

Script Output | Query Result

SQL | Fetched 50 rows in 0.026 seconds

	EXERCISE_ID	EXERCISE_HOURS_PER_WEEK	BMI	PHYSICAL_ACTIVITY_DAYS_PER_WEEK	SLEEP_HOURS_PER_DAY	PATIENT_ID
1	1	8.123736391	27.09844517	3	5	AAA9246
2	2	14.71001101	32.41282532	1	7	AAC3641
3	3	6.935790631	28.21990339	6	6	AAD0466
4	4	4.885204408	33.29690064	5	6	AAD3011
5	5	16.56956856	24.1210454	2	5	AAE4089
6	6	13.65483713	35.1629374	7	8	AAF0810
7	7	0.675423373	32.78469285	1	8	AAL7171
8	8	8.129950576	24.73609069	2	10	AAM1257
9	9	7.815475337	35.986794	0	5	AAM2719
10	10	18.82658048	28.70674772	4	9	AAM9231
11	11	7.304884549	30.48437416	4	9	AAN5057
12	12	7.588174044	25.49004109	3	10	AAO6684

Declaration"

5. QUESTIONS

Finally, respond to the questions proposed by your group:

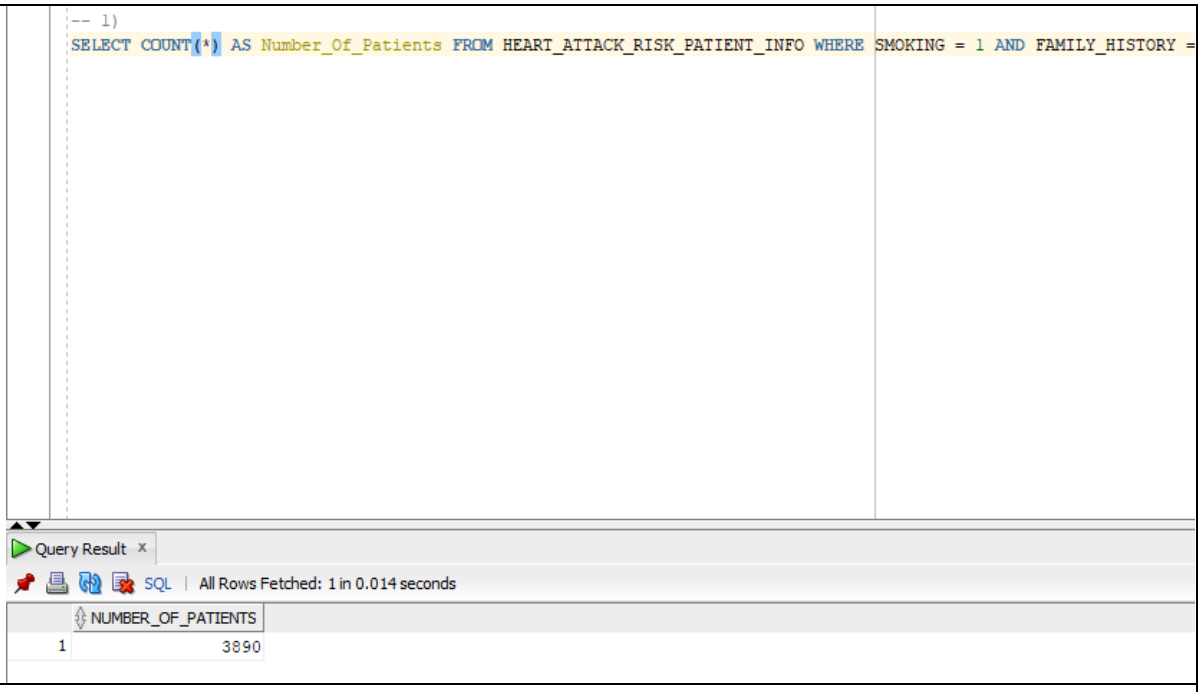
- Each question MUST be answered with data available in the proposed normalized database.
- However, extra credit will be assigned if you also present an equivalent query using the initial raw table data created using SQLLDR.

Question 1	**List the number of patients who smoke and have a family history of heart problems.**

SQL	`SELECT COUNT(*) AS Number_of_Patients`
	`FROM HARPINFO_PATIENT`
	`WHERE SMOKING = 1 AND FAMILY_HISTORY = 1;`

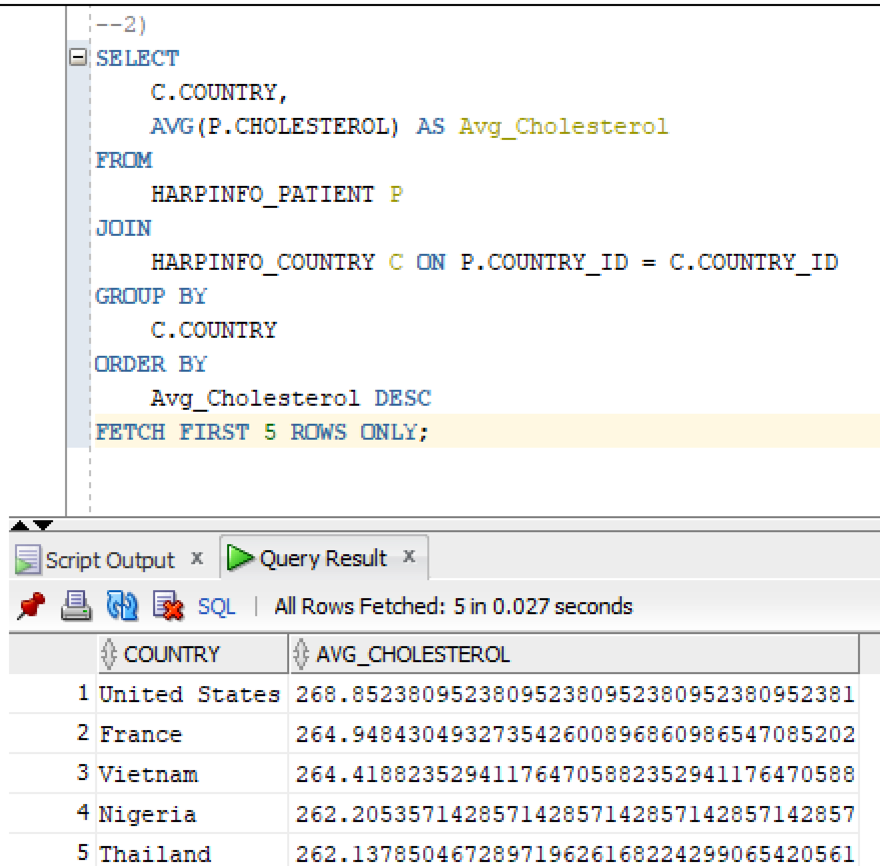
Result	`-- 1)`
	`SELECT COUNT(*) AS Number_of_Patients`
	`FROM HARPINFO_PATIENT`
	`WHERE SMOKING = 1 AND FAMILY_HISTORY = 1;`
	Script Output | Query Result
	SQL | All Rows Fetched: 1 in 0.017 seconds
	NUMBER_OF_PATIENTS
	1 3890

SQL (raw) [optional]	`SELECT COUNT(*) AS Number_Of_Patients`
	`FROM HEART_ATTACK_RISK_PATIENT_INFO`
	`WHERE SMOKING = 1 AND FAMILY_HISTORY = 1;`

Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 2	**List the top 5 countries with the highest average cholesterol levels.**

SQL	`SELECT`
	`  C.COUNTRY,`
	`  AVG(P.CHOLESTEROL) AS Avg_Cholesterol`
	`FROM`
	`  HARPINFO_PATIENT P`
	`JOIN`
	`  HARPINFO_COUNTRY C ON P.COUNTRY_ID = C.COUNTRY_ID`
	`GROUP BY`
	`  C.COUNTRY`
	`ORDER BY`
	`  Avg_Cholesterol DESC`
	`FETCH FIRST 5 ROWS ONLY;`

<table>
<tr><td>Result</td><td>


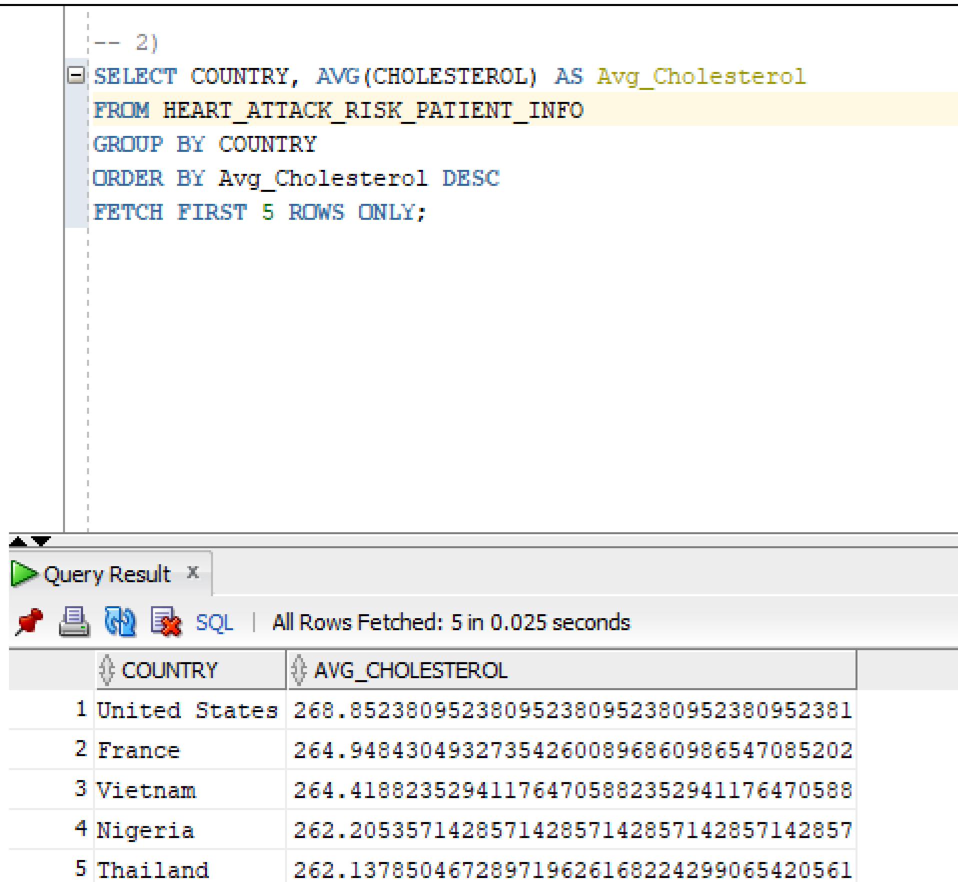
```
--2)
SELECT
    C.COUNTRY,
    AVG(P.CHOLESTEROL) AS Avg_Cholesterol
FROM
    HARPINFO_PATIENT P
JOIN
    HARPINFO_COUNTRY C ON P.COUNTRY_ID = C.COUNTRY_ID
GROUP BY
    C.COUNTRY
ORDER BY
    Avg_Cholesterol DESC
FETCH FIRST 5 ROWS ONLY;
```

Script Output × | Query Result ×

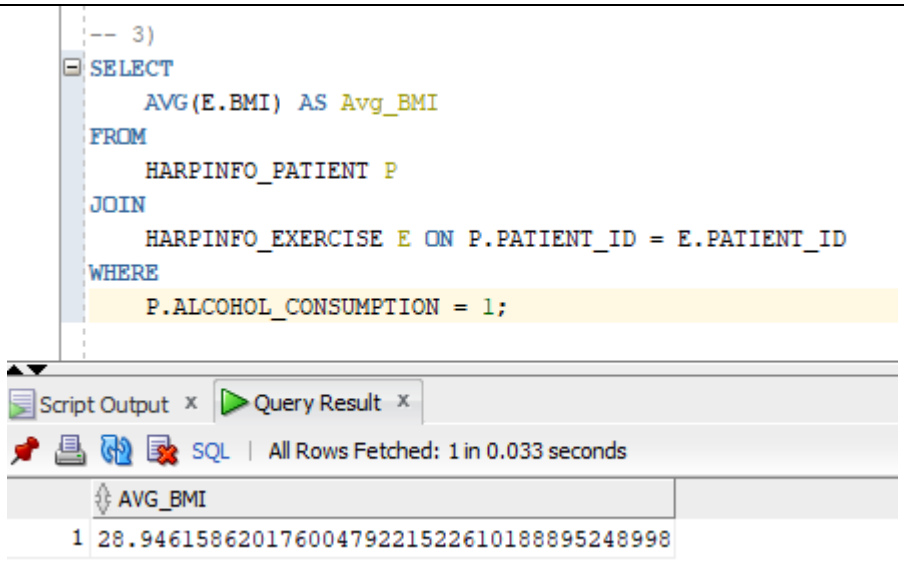
SQL | All Rows Fetched: 5 in 0.027 seconds

	COUNTRY	AVG_CHOLESTEROL
1	United States	268.8523809523809523809523809523809523 81
2	France	264.9484304932735426008968609865470852 02
3	Vietnam	264.4188235294117647058823529411764705 88
4	Nigeria	262.2053571428571428571428571428571428 57
5	Thailand	262.1378504672897196261682242990654205 61
</td></tr>
<tr><td>SQL (raw) [optional]</td><td>

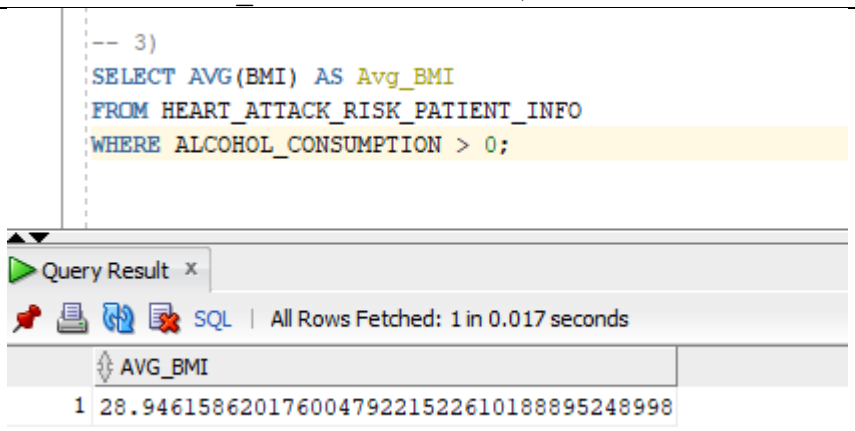
```
SELECT COUNTRY, AVG(CHOLESTEROL) AS Avg_Cholesterol
FROM HEART_ATTACK_RISK_PATIENT_INFO
GROUP BY COUNTRY
ORDER BY Avg_Cholesterol DESC
FETCH FIRST 5 ROWS ONLY;
```
</td></tr>
</table>

<table>
<tr><td>Result (raw)</td><td> 
<pre>
-- 2)
SELECT COUNTRY, AVG(CHOLESTEROL) AS Avg_Cholesterol
FROM HEART_ATTACK_RISK_PATIENT_INFO
GROUP BY COUNTRY
ORDER BY Avg_Cholesterol DESC
FETCH FIRST 5 ROWS ONLY;
</pre>
Query Result
SQL | All Rows Fetched: 5 in 0.025 seconds
<table>
<tr><th></th><th>COUNTRY</th><th>AVG_CHOLESTEROL</th></tr>
<tr><td>1</td><td>United States</td><td>268.8523809523809523809523809523809523810952381</td></tr>
<tr><td>2</td><td>France</td><td>264.9484304932735426008968609865470852O2</td></tr>
<tr><td>3</td><td>Vietnam</td><td>264.4188235294117647058823529411764705 88</td></tr>
<tr><td>4</td><td>Nigeria</td><td>262.2053571428571428571428571428571428 57</td></tr>
<tr><td>5</td><td>Thailand</td><td>262.1378504672897196261682242990654205 61</td></tr>
</table>
</td></tr>
<tr><td>Comments</td><td>The outputs in both cases appear to be the same</td></tr>
</table>

<table>
<tr><td>Question 3</td><td>What is the average BMI for patients who consume alcohol?</td></tr>
<tr><td>SQL</td><td>
<pre>
SELECT
    AVG(E.BMI) AS Avg_BMI
FROM
    HARPINFO_PATIENT P
JOIN
    HARPINFO_EXERCISE E ON P.PATIENT_ID = E.PATIENT_ID
WHERE
    P.ALCOHOL_CONSUMPTION = 1;
</pre>
</td></tr>
</table>

Result	

SQL (raw) [optional]	`SELECT AVG(BMI) AS Avg_BMI`
	`FROM HEART_ATTACK_RISK_PATIENT_INFO`
	`WHERE ALCOHOL_CONSUMPTION > 0;`

Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 4	**Is there a relationship between physical activity of patients and obesity?**

SQL	`SELECT`
	`  CASE`
	`  WHEN P.OBESITY = 1 THEN 'Obese'`
	`  ELSE 'Non-Obese'`
	`  END AS Obesity_Status,`
	`  AVG(E.EXERCISE_HOURS_PER_WEEK) AS Avg_Exercise_Hours_Per_Week,`
	`  AVG(E.PHYSICAL_ACTIVITY_DAYS_PER_WEEK) AS Avg_Physical_Activity_Days_Per_Week`
	`FROM`
	`  HARPINFO_PATIENT P`
	`JOIN`

```
    HARPINFO_EXERCISE E ON P.PATIENT_ID = E.PATIENT_ID
GROUP BY
    P.OBESITY;
```

Result

```
-- 4)
SELECT
    CASE
        WHEN P.OBESITY = 1 THEN 'Obese'
        ELSE 'Non-Obese'
    END AS Obesity_Status,
    AVG(E.EXERCISE_HOURS_PER_WEEK) AS Avg_Exercise_Hours_Per_Week,
    AVG(E.PHYSICAL_ACTIVITY_DAYS_PER_WEEK) AS Avg_Physical_Activity_Days_Per_Week
FROM
    HARPINFO_PATIENT P
JOIN
    HARPINFO_EXERCISE E ON P.PATIENT_ID = E.PATIENT_ID
GROUP BY
    P.OBESITY;
```

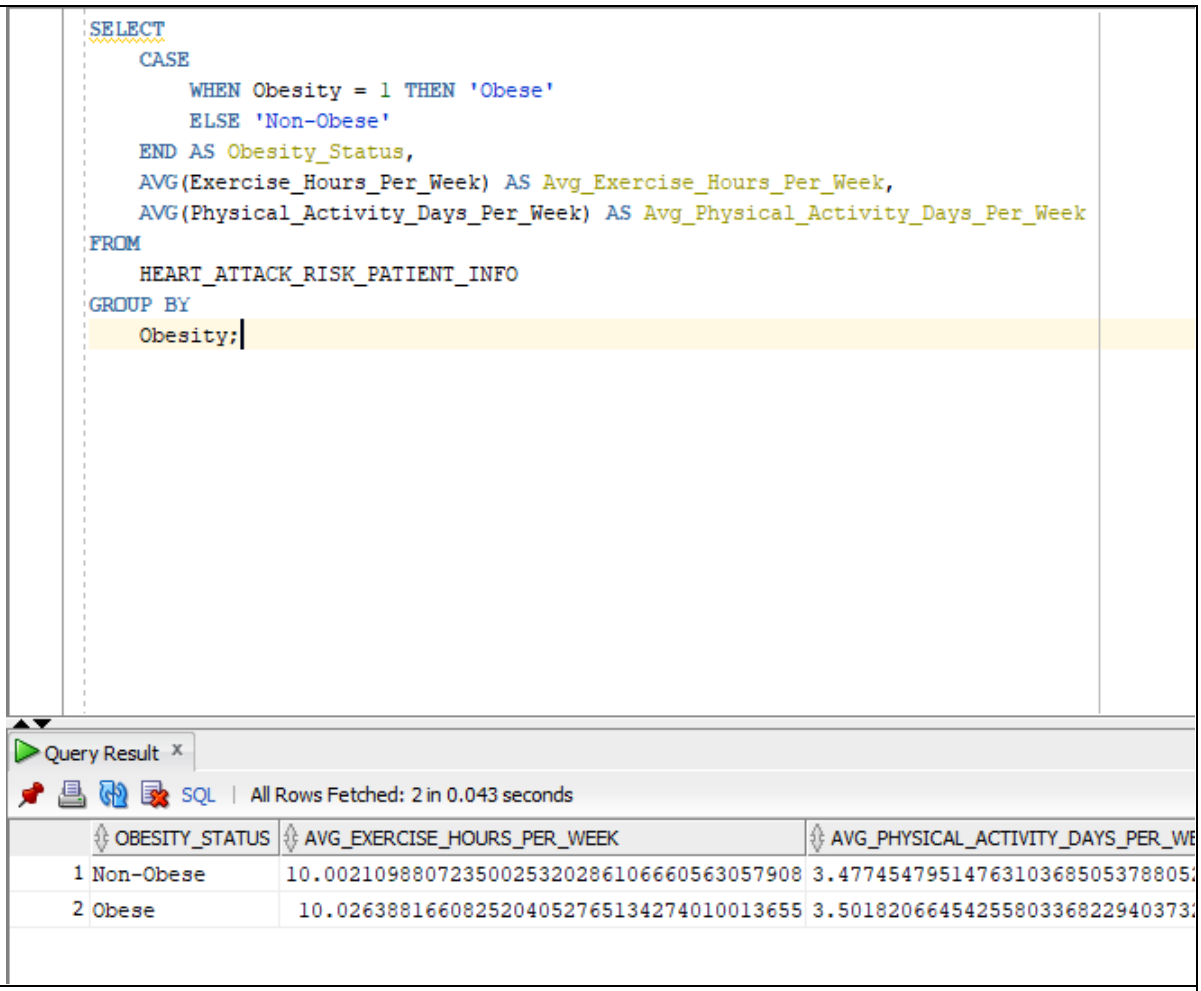
Script Output | Query Result

SQL | All Rows Fetched: 2 in 0.036 seconds

	OBESITY_STATUS	AVG_EXERCISE_HOURS_PER_WEEK	AVG_PHYSICAL_ACTIVITY_DAYS_PER_
1	Non-Obese	10.002109880723500253202861066605630579 08	3.477454795147631036850537880
2	Obese	10.026388166082520405276513427401001365 5	3.501820664542558033682294037

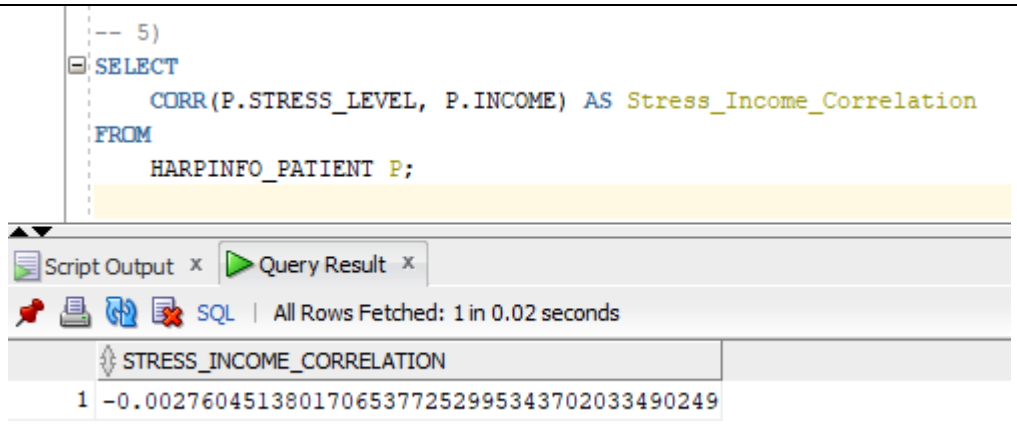
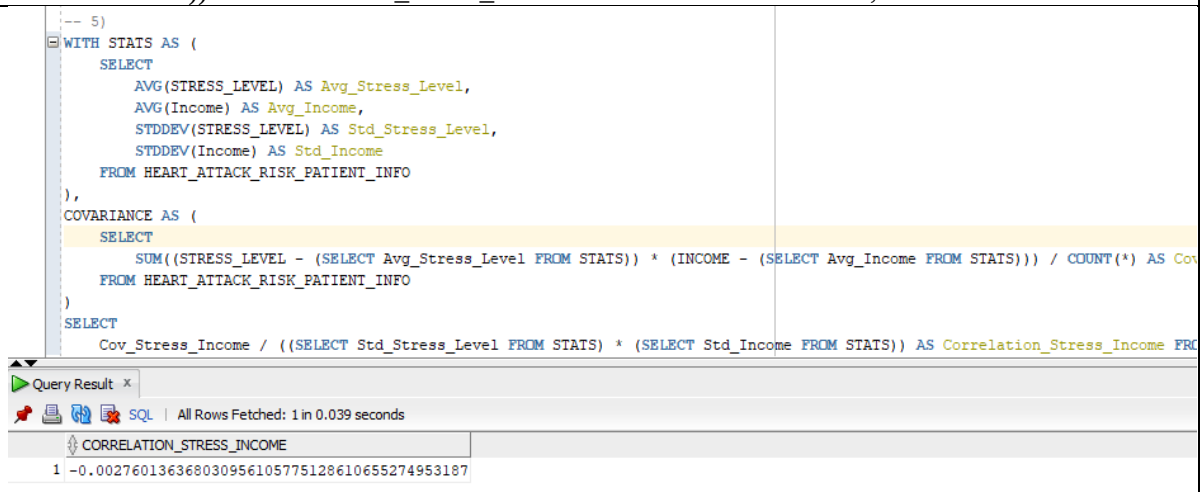
SQL (raw) [optional]

```
SELECT
   CASE
      WHEN Obesity = 1 THEN 'Obese'
      ELSE 'Non-Obese'
   END AS Obesity_Status,
   AVG(Exercise_Hours_Per_Week) AS Avg_Exercise_Hours_Per_Week,
   AVG(Physical_Activity_Days_Per_Week) AS Avg_Physical_Activity_Days_Per_Week
FROM
   HEART_ATTACK_RISK_PATIENT_INFO
GROUP BY
   Obesity;
```

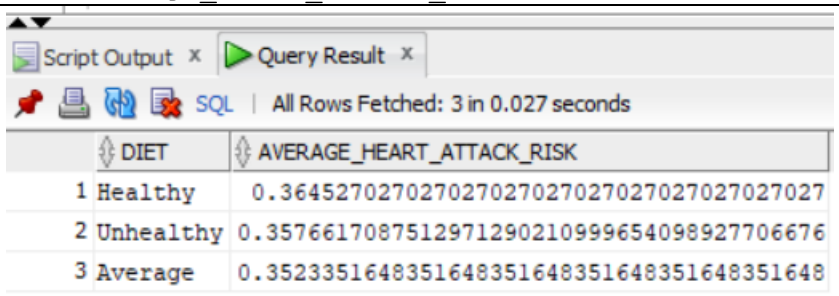
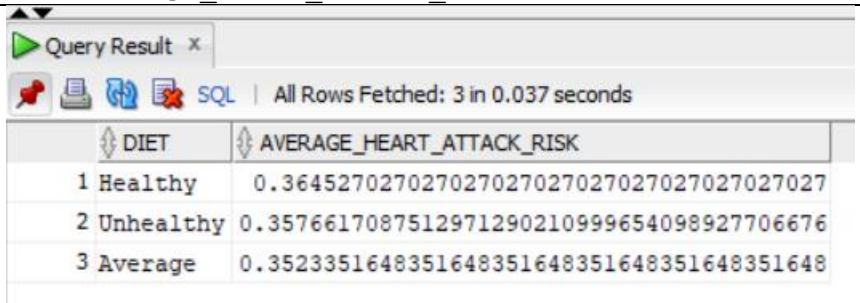
Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 5	**Is there a correlation between stress level and income level of patients?**

SQL	`SELECT`
	`  CORR(P.STRESS_LEVEL, P.INCOME) AS Stress_Income_Correlation`
	`FROM`
	`  HARPINFO_PATIENT P;`

Result	
SQL (raw) [optional]	WITH STATS AS (SELECT AVG(STRESS_LEVEL) AS Avg_Stress_Level, AVG(Income) AS Avg_Income, STDDEV(STRESS_LEVEL) AS Std_Stress_Level, STDDEV(Income) AS Std_Income FROM HEART_ATTACK_RISK_PATIENT_INFO), COVARIANCE AS (SELECT SUM((STRESS_LEVEL - (SELECT Avg_Stress_Level FROM STATS)) * (INCOME - (SELECT Avg_Income FROM STATS))) / COUNT(*) AS Cov_Stress_Income FROM HEART_ATTACK_RISK_PATIENT_INFO) SELECT Cov_Stress_Income / ((SELECT Std_Stress_Level FROM STATS) * (SELECT Std_Income FROM STATS)) AS Correlation_Stress_Income FROM COVARIANCE;
Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 6	**Is there a correlation between dietary habits of patients and their heart**

	attack risks?
SQL	`SELECT DH.DIET AS Diet, AVG(P.HEART_ATTACK_RISK) AS Average_Heart_Attack_Risk FROM HARPINFO_PATIENT P JOIN HARPINFO_DIETARY_HABIT DH ON P.DIET_ID = DH.DIET_ID GROUP BY DH.DIET ORDER BY Average_Heart_Attack_Risk DESC;`
Result	 Script Output × Query Result × SQL | All Rows Fetched: 3 in 0.027 seconds DIET | AVERAGE_HEART_ATTACK_RISK 1 Healthy 0.364527027027027027027027027027027027027 2 Unhealthy 0.35766170875129712902109996540989277066 76 3 Average 0.352335164835164835164835164835164835164 8
SQL (raw) [optional]	`SELECT Diet, AVG(Heart_Attack_Risk) AS Average_Heart_Attack_Risk FROM HEART_ATTACK_RISK_PATIENT_INFO GROUP BY Diet ORDER BY Average_Heart_Attack_Risk DESC;`
Result (raw)	 Query Result × SQL | All Rows Fetched: 3 in 0.037 seconds DIET | AVERAGE_HEART_ATTACK_RISK 1 Healthy 0.364527027027027027027027027027027027027 2 Unhealthy 0.35766170875129712902109996540989277066 76 3 Average 0.352335164835164835164835164835164835164 8
Comments	The outputs in both cases appear to be the same

Question 7	**List the countries where more than 25% of the patients have a heart attack risk**
SQL	`SELECT C.COUNTRY AS Country, COUNT(P.PATIENT_ID) AS Total_Patients, SUM(CASE WHEN P.HEART_ATTACK_RISK = 1 THEN 1 ELSE 0 END) AS Patients_With_Heart_Attack_Risk, (SUM(CASE WHEN P.HEART_ATTACK_RISK = 1 THEN 1 ELSE 0 END) *`

```
100.0 / COUNT(P.PATIENT_ID)) AS Heart_Attack_Risk_Percentage
FROM
    HARPINFO_PATIENT P
JOIN
    HARPINFO_COUNTRY C ON P.COUNTRY_ID = C.COUNTRY_ID
GROUP BY
    C.COUNTRY
HAVING
    (SUM(CASE WHEN P.HEART_ATTACK_RISK = 1 THEN 1 ELSE 0 END) *
100.0 / COUNT(P.PATIENT_ID)) > 25
ORDER BY
    Heart_Attack_Risk_Percentage DESC;
```

Result

Script Output | Query Result

SQL | All Rows Fetched: 20 in 0.034 seconds

	COUNTRY	TOTAL_PATIENTS	PATIENTS_WITH_HEART_ATTACK_RISK	HEART_ATTACK_RISK_PERCENTAGE
1	South Korea	409	163	39.85330073349633251833740831
2	Nigeria	448	178	39.73214285714285714285714285
3	United States	420	166	39.52380952380952380952380952
4	Colombia	429	162	37.76223776223776223776223776
5	Thailand	428	161	37.61682242990654205607476635
6	Australia	449	168	37.41648106904231625835189309
7	Argentina	471	174	36.94267515923566878980891719
8	Germany	477	172	36.05870020964360587002096436
9	Canada	440	158	35.90909090909090909090909090
10	China	436	155	35.55045871559633027522935779
11	Brazil	462	163	35.28138528138528138528138528
12	France	446	157	35.20179372197309417040358744
13	United Kingdom	457	160	35.01094091903719912472647702
14	Spain	430	150	34.88372093023255813953488372
15	Vietnam	425	148	34.82352941176470588235294117
16	New Zealand	435	151	34.71264367816091954022988505
17	South Africa	425	144	33.88235294117647058823529411
18	Japan	433	144	33.25635103926096997690531177
19	Italy	431	136	31.55452436194895591647331786
20	India	412	129	31.31067961165048543689320388

SQL (raw) [optional]

```
SELECT
    Country,
    COUNT(*) AS Total_Patients,
    SUM(CASE WHEN Heart_Attack_Risk = 1 THEN 1 ELSE 0 END) AS
Patients_With_Heart_Attack_Risk,
    (SUM(CASE WHEN Heart_Attack_Risk = 1 THEN 1 ELSE 0 END) * 100.0
/ COUNT(*)) AS Heart_Attack_Risk_Percentage
FROM
    HEART_ATTACK_RISK_PATIENT_INFO
GROUP BY
```

```
    Country
HAVING
    (SUM(CASE WHEN Heart_Attack_Risk = 1 THEN 1 ELSE 0 END) * 100.0
/ COUNT(*)) > 25
ORDER BY
    Heart_Attack_Risk_Percentage DESC;
```

Result (raw)

Query Result | SQL | All Rows Fetched: 20 in 0.029 seconds

	COUNTRY	TOTAL_PATIENTS	PATIENTS_WITH_HEART_ATTACK_RISK	HEART_ATTACK_RISK_PERCENTAGE
1	South Korea	409	163	39.85330073349633251833740831291
2	Nigeria	448	178	39.73214285714285714285714285714
3	United States	420	166	39.52380952380952380952380952381
4	Colombia	429	162	37.76223776223776223776223776224
5	Thailand	428	161	37.61682242990654205607476635514
6	Australia	449	168	37.41648106904231625835189309577
7	Argentina	471	174	36.94267515923566878980891719745
8	Germany	477	172	36.05870020964360587002096436059
9	Canada	440	158	35.90909090909090909090909090909
10	China	436	155	35.55045871559633027522935779817
11	Brazil	462	163	35.28138528138528138528138528139
12	France	446	157	35.20179372197309417040358744395
13	United Kingdom	457	160	35.01094091903719912472647702407
14	Spain	430	150	34.88372093023255813953488372093
15	Vietnam	425	148	34.82352941176470588235294117647
16	New Zealand	435	151	34.71264367816091954022988505747
17	South Africa	425	144	33.88235294117647058823529411765
18	Japan	433	144	33.25635103926096997690531177829
19	Italy	431	136	31.55452436194895591647331786543
20	India	412	129	31.31067961165048543689320388350

Comments: The outputs in both cases appear to be the same

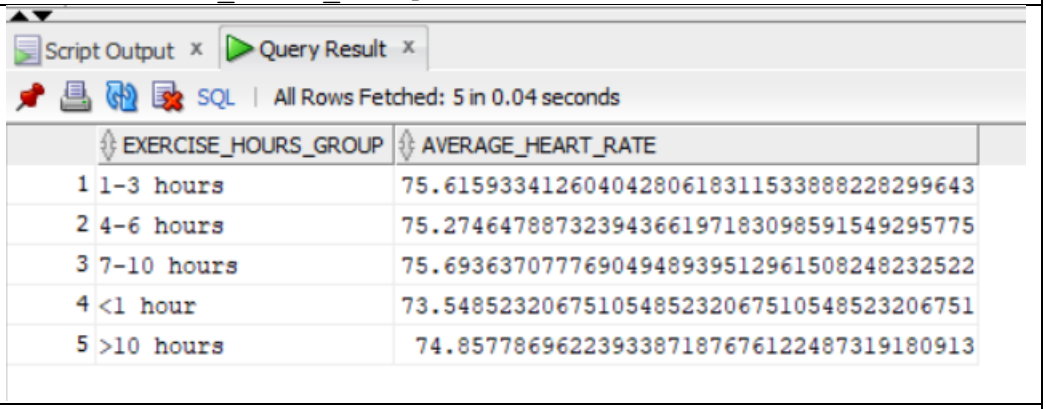
Question 8	**How does the number of exercise hours per week relate to heart rates among patients?**

SQL

```
SELECT
    CASE
        WHEN E.EXERCISE_HOURS_PER_WEEK < 1 THEN '<1 hour'
        WHEN E.EXERCISE_HOURS_PER_WEEK BETWEEN 1 AND 3 THEN
'1-3 hours'
        WHEN E.EXERCISE_HOURS_PER_WEEK BETWEEN 4 AND 6 THEN
'4-6 hours'
        WHEN  E.EXERCISE_HOURS_PER_WEEK  BETWEEN  7  AND  10
```

```
THEN '7-10 hours'
        ELSE '>10 hours'
    END AS Exercise_Hours_Group,
    AVG(P.HEART_RATE) AS Average_Heart_Rate
FROM
    HARPINFO_EXERCISE E
JOIN
    HARPINFO_PATIENT P ON E.PATIENT_ID = P.PATIENT_ID
GROUP BY
    CASE
        WHEN E.EXERCISE_HOURS_PER_WEEK < 1 THEN '<1 hour'
        WHEN E.EXERCISE_HOURS_PER_WEEK BETWEEN 1 AND 3 THEN
'1-3 hours'
        WHEN E.EXERCISE_HOURS_PER_WEEK BETWEEN 4 AND 6 THEN
'4-6 hours'
        WHEN  E.EXERCISE_HOURS_PER_WEEK  BETWEEN  7  AND  10
THEN '7-10 hours'
        ELSE '>10 hours'
    END
ORDER BY
    Exercise_Hours_Group;
```

Result



	EXERCISE_HOURS_GROUP	AVERAGE_HEART_RATE
1	1-3 hours	75.61593341260404280618311533888228299643
2	4-6 hours	75.27464788732394366197183098591549295775
3	7-10 hours	75.69363707776904948939512961508248232522
4	<1 hour	73.54852320675105485232067510548523206751
5	>10 hours	74.85778696223933871876761224873191809l3

SQL (raw) [optional]

```
SELECT
    CASE
        WHEN Exercise_Hours_Per_Week < 1 THEN '<1 hour'
        WHEN Exercise_Hours_Per_Week BETWEEN 1 AND 3 THEN
'1-3 hours'
        WHEN Exercise_Hours_Per_Week BETWEEN 4 AND 6 THEN
'4-6 hours'
        WHEN Exercise_Hours_Per_Week BETWEEN 7 AND 10 THEN
'7-10 hours'
        ELSE '>10 hours'
    END AS Exercise_Hours_Group,
    AVG(Heart_Rate) AS Average_Heart_Rate
FROM
    HEART_ATTACK_RISK_PATIENT_INFO
GROUP BY
    CASE
        WHEN Exercise_Hours_Per_Week < 1 THEN '<1 hour'
```

```
        WHEN Exercise_Hours_Per_Week BETWEEN 1 AND 3 THEN
'1-3 hours'
        WHEN Exercise_Hours_Per_Week BETWEEN 4 AND 6 THEN
'4-6 hours'
        WHEN Exercise_Hours_Per_Week BETWEEN 7 AND 10 THEN
'7-10 hours'
        ELSE '>10 hours'
    END
ORDER BY
    Exercise_Hours_Group;
```

Result (raw)

Query Result | SQL | All Rows Fetched: 5 in 0.033 seconds

	EXERCISE_HOURS_GROUP	AVERAGE_HEART_RATE
1	1-3 hours	75.61593341260404280618311533888228299643
2	4-6 hours	75.27464788732394366197183098591549295775
3	7-10 hours	75.69363707776904948939512961508248232522
4	<1 hour	73.54852320675105485232067510548523206751
5	>10 hours	74.85778696223933871876761224873191809 13

Comments: The outputs in both cases appear to be the same

Question 9: **Determine the number of patients who have previous heart problems grouped by their gender?**

SQL

```
SELECT
    P.SEX AS Gender,
    COUNT(P.PATIENT_ID)                                    AS
Number_of_Patients_With_Heart_Problems
FROM
    HARPINFO_PATIENT P
WHERE
    P.PREVIOUS_HEART_PROBLEMS = 1
GROUP BY
    P.SEX
ORDER BY
    Number_of_Patients_With_Heart_Problems DESC;
```

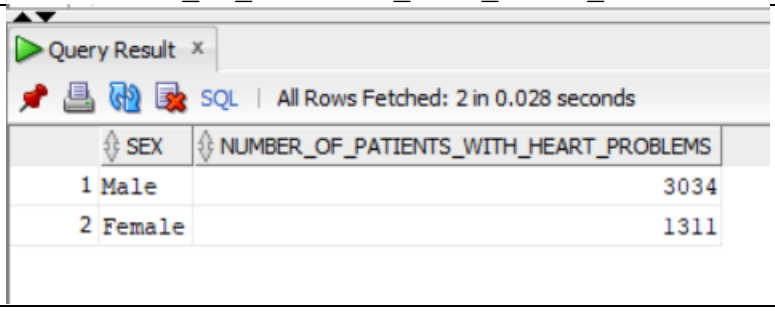
Result

Script Output | Query Result | SQL | All Rows Fetched: 2 in 0.026 seconds

	GENDER	NUMBER_OF_PATIENTS_WITH_HEART_PROBLEMS
1	Male	3034
2	Female	1311

SQL (raw) [optional]

```
SELECT
    Sex,
    COUNT(*) AS Number_of_Patients_With_Heart_Problems
FROM
```

	HEART_ATTACK_RISK_PATIENT_INFO WHERE Previous_Heart_Problems = 1 GROUP BY Sex ORDER BY Number_of_Patients_With_Heart_Problems DESC;
Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 10	**Find the average number of exercise hours per week for patients from different countries.**
SQL	SELECT C.COUNTRY AS Country, AVG(E.EXERCISE_HOURS_PER_WEEK) AS Average_Exercise_Hours_Per_Week FROM HARPINFO_EXERCISE E JOIN HARPINFO_PATIENT P ON E.PATIENT_ID = P.PATIENT_ID JOIN HARPINFO_COUNTRY C ON P.COUNTRY_ID = C.COUNTRY_ID GROUP BY C.COUNTRY ORDER BY Average_Exercise_Hours_Per_Week DESC;

Result

Script Output × | Query Result ×

SQL | All Rows Fetched: 20 in 0.044 seconds

	COUNTRY	AVERAGE_EXERCISE_HOURS_PER_WEEK
1	Nigeria	10.37975783003348214285714285714285714286
2	Germany	10.28210022156603773584905660377358490566
3	New Zealand	10.27001718632873563218390804597701149425
4	Italy	10.12801526217401392111368909512761020882
5	United States	10.12658161927857142857142857142857142857
6	China	10.11154680961926605504587155963302752294
7	India	10.10979229442961165048543689320388349515
8	Brazil	10.10322730016233766233766233766233766234
9	Canada	10.08751929862045454545454545454545454545
10	Vietnam	10.08209232351058823529411764705882352941
11	South Africa	10.03916698449882352941176470588235294118
12	United Kingdom	10.03572597194967177242888402625820568928
13	France	9.9932674979573991031390134529147982062 8
14	South Korea	9.9642981972713936430317848410757946210 3
15	Colombia	9.8736417835990675990675990675990675990 7
16	Argentina	9.8613627877940552016985138004246284501 1
17	Japan	9.8544768608498845265588914549653579676 7
18	Spain	9.8279660351
19	Thailand	9.6627221106191588785046728971962616822 4
20	Australia	9.4770585859576837416481069042316258351 9

SQL (raw) [optional]

```
SELECT
    Country,
    AVG(Exercise_Hours_Per_Week)                                    AS
Average_Exercise_Hours_Per_Week
FROM
    HEART_ATTACK_RISK_PATIENT_INFO
GROUP BY
    Country
ORDER BY
    Average_Exercise_Hours_Per_Week DESC;
```

Result (raw)

Script Output × | Query Result ×

SQL | All Rows Fetched: 20 in 0.027 seconds

	COUNTRY	AVERAGE_EXERCISE_HOURS_PER_WEEK
1	Nigeria	10.379757830033482142857142857142857142 86
2	Germany	10.28210022156603773584905660377358490566
3	New Zealand	10.27001718632873563218390804597701149425
4	Italy	10.12801526217401392111368909512761020882
5	United States	10.12658161927857142857142857142857142857
6	China	10.11154680961926605504587155963302752294
7	India	10.10979229442961165048543689320388349515
8	Brazil	10.10322730016233766233766233766233766234
9	Canada	10.08751929862045454545454545454545454545
10	Vietnam	10.08209232351058823529411764705882352941
11	South Africa	10.03916698449882352941176470588235294118
12	United Kingdom	10.03572597194967177242888402625820568928
13	France	9.9932674979573991031390134529147982062 8
14	South Korea	9.9642981972713936430317848410757946210 3
15	Colombia	9.8736417835990675990675990675990675990 7
16	Argentina	9.8613627877940552016985138004246284501 1
17	Japan	9.8544768608498845265588914549653579676 7
18	Spain	9.8279660351
19	Thailand	9.6627221106191588785046728971962616822 4
20	Australia	9.4770585859576837416481069042316258351 9

Comments	The outputs in both cases appear to be the same

Question 11	**What is the average age of patients who have a heart attack risk?**

SQL	`SELECT AVG(P.AGE) AS Average_Age`
	`FROM HARPINFO_PATIENT P`
	`WHERE P.HEART_ATTACK_RISK = 1;`

Result	(see table below)

	AVERAGE_AGE
1	53.89009238611022618668365721567378145906

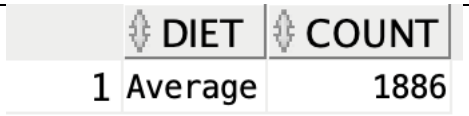
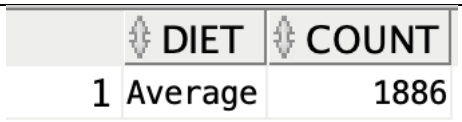
SQL (raw) [optional]	`SELECT AVG(Age) AS Average_Age`
	`FROM HEART_ATTACK_RISK_PATIENT_INFO`
	`WHERE Heart_Attack_Risk = 1;`

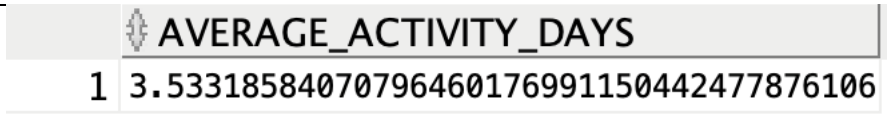
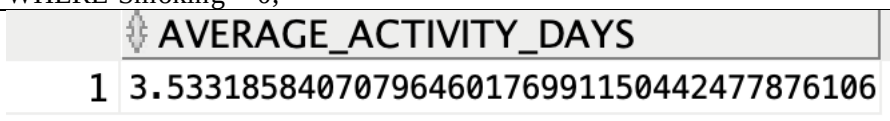
Result (raw)	(see table below)

	AVERAGE_AGE
1	53.89009238611022618668365721567378145906

Comments	The outputs in both cases appear to be the same

Question 12	**What is the most common diet type among patients with a low heart attack risk?**

SQL	SELECT DH.DIET AS Diet, COUNT(*) AS Count FROM HARPINFO_PATIENT P JOIN HARPINFO_DIETARY_HABIT DH ON P.DIET_ID = DH.DIET_ID WHERE P.HEART_ATTACK_RISK = 0 GROUP BY DH.DIET ORDER BY Count DESC FETCH FIRST 1 ROW ONLY;
Result	
SQL (raw) [optional]	SELECT Diet, COUNT(*) AS Count FROM HEART_ATTACK_RISK_PATIENT_INFO WHERE Heart_Attack_Risk = 0 GROUP BY Diet ORDER BY Count DESC FETCH FIRST 1 ROW ONLY;
Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 13	**What is the average number of physical activity days per week for patients who do not smoke?**
SQL	SELECT AVG(E.PHYSICAL_ACTIVITY_DAYS_PER_WEEK) AS Average_Activity_Days FROM HARPINFO_EXERCISE E JOIN HARPINFO_PATIENT P ON E.PATIENT_ID = P.PATIENT_ID WHERE P.SMOKING = 0;
Result	
SQL (raw) [optional]	SELECT AVG(Physical_Activity_Days_Per_Week) AS Average_Activity_Days FROM HEART_ATTACK_RISK_PATIENT_INFO WHERE Smoking = 0;
Result (raw)	
Comments	The outputs in both cases appear to be the same

Question 14	What is the average age of patients with high cholesterol levels who have a heart attack risk in each country?
SQL	SELECT C.COUNTRY AS Country, AVG(P.AGE) AS Average_Age FROM HARPINFO_PATIENT P JOIN HARPINFO_COUNTRY C ON P.COUNTRY_ID = C.COUNTRY_ID WHERE P.HEART_ATTACK_RISK = 1 AND P.CHOLESTEROL >= 240 GROUP BY C.COUNTRY

ORDER BY Average_Age DESC;

Result

	COUNTRY	AVERAGE_AGE
1	South Korea	58.956043956043956043956043956043956043956
2	New Zealand	56.807692307692307692307692307692307692307
3	India	56.528571428571428571428571428571428571428
4	United Kingdom	56.473684210526315789473684210526315789473
5	Italy	56.298850574712643678160919540229885057471
6	Canada	55.876288659793814432989690721649484536082
7	Argentina	55.575471698113207547169811320754716981132
8	Vietnam	54.581395348837209302325581395348837209302
9	Japan	54.505747126436781609195402298850574712643
10	United States	54.242718446601941747572815533980582524271
11	South Africa	54.223529411764705882352941176470588235294
12	Thailand	54.077669902912621359223300970873786407766
13	Nigeria	53.911764705882352941176470588235294117647
14	France	53.835051546391752577319587628865979381443
15	Spain	53.552941176470588235294117647058823529411
16	Colombia	52.593406593406593406593406593406593406593
17	China	52.3125
18	Australia	52.073684210526315789473684210526315789473
19	Brazil	49.144444444444444444444444444444444444444
20	Germany	47.93

SQL (raw) [optional]

```
SELECT
   Country,
   AVG(Age) AS Average_Age
FROM
   HEART_ATTACK_RISK_PATIENT_INFO
WHERE
   Heart_Attack_Risk = 1
   AND Cholesterol >= 240
GROUP BY
   Country
ORDER BY
   Average_Age DESC
```

Result (raw)	<table><tr><th></th><th>COUNTRY</th><th>AVERAGE_AGE</th></tr><tr><td>1</td><td>South Korea</td><td>58.95604395604395604395604395604396</td></tr><tr><td>2</td><td>New Zealand</td><td>56.80769230769230769230769230769231</td></tr><tr><td>3</td><td>India</td><td>56.52857142857142857142857142857143</td></tr><tr><td>4</td><td>United Kingdom</td><td>56.47368421052631578947368421052631578947</td></tr><tr><td>5</td><td>Italy</td><td>56.29885057471264367816091954022988505747</td></tr><tr><td>6</td><td>Canada</td><td>55.87628865979381443298969072164948453608</td></tr><tr><td>7</td><td>Argentina</td><td>55.57547169811320754716981132075471698113</td></tr><tr><td>8</td><td>Vietnam</td><td>54.5813953488372093023255813953488372093</td></tr><tr><td>9</td><td>Japan</td><td>54.50574712643678160919540229885057471264</td></tr><tr><td>10</td><td>United States</td><td>54.24271844660194174757281553398058252427</td></tr><tr><td>11</td><td>South Africa</td><td>54.22352941176470588235294117647058823529</td></tr><tr><td>12</td><td>Thailand</td><td>54.07766990291262135922330097087378640777</td></tr><tr><td>13</td><td>Nigeria</td><td>53.91176470588235294117647058823529411765</td></tr><tr><td>14</td><td>France</td><td>53.83505154639175257731958762886597938144</td></tr><tr><td>15</td><td>Spain</td><td>53.55294117647058823529411764705882352941</td></tr><tr><td>16</td><td>Colombia</td><td>52.59340659340659340659340659340659340659</td></tr><tr><td>17</td><td>China</td><td>52.3125</td></tr><tr><td>18</td><td>Australia</td><td>52.07368421052631578947368421052631578947</td></tr><tr><td>19</td><td>Brazil</td><td>49.14444444444444444444444444444444</td></tr><tr><td>20</td><td>Germany</td><td>47.93</td></tr></table>
Comments	The outputs in both cases appear to be the same

Question 15	**What is the percentage of patients with a family history of heart problems who have a high cholesterol level (cholesterol >= 240) compared to those without a family history of heart problems?**

SQL	<pre>WITH Family_History_Stats AS (
	    SELECT
	        CASE
	            WHEN P.FAMILY_HISTORY = 1 THEN 'With Family History'
	            ELSE 'Without Family History'
	        END AS Family_History_Status,
	        SUM(CASE WHEN P.CHOLESTEROL >= 240 THEN 1 ELSE 0 END) AS High_Cholesterol_Count,
	        COUNT(P.PATIENT_ID) AS Total_Count</pre>

```
    FROM
        HARPINFO_PATIENT P
    GROUP BY
        CASE
            WHEN P.FAMILY_HISTORY = 1 THEN 'With Family History'
            ELSE 'Without Family History'
        END
)
SELECT
    Family_History_Status,
    High_Cholesterol_Count,
    Total_Count,
    (High_Cholesterol_Count    *    100.0    /    Total_Count)    AS
Percentage_High_Cholesterol
FROM
    Family_History_Stats
ORDER BY
    Family_History_Status
```

Result

	FAMILY_HISTORY_STATUS	HIGH_CHOLESTEROL_COUNT	TOTAL_COUNT	PERCENTAGE_HIGH_CHOLI
1	With Family History	2454	4320	56.805555555555555555555555
2	Without Family History	2600	4443	58.519018681071348188161152

SQL (raw) [optional]

```
WITH Family_History_Stats AS (
   SELECT
      CASE
         WHEN Family_History = 1 THEN 'With Family History'
         ELSE 'Without Family History'
      END AS Family_History_Status,
      SUM(CASE WHEN Cholesterol >= 240 THEN 1 ELSE 0 END) AS High_Cholesterol_Count,
      COUNT(*) AS Total_Count
   FROM
      HEART_ATTACK_RISK_PATIENT_INFO
   GROUP BY
      CASE
         WHEN Family_History = 1 THEN 'With Family History'
         ELSE 'Without Family History'
      END
)
SELECT
   Family_History_Status,
   High_Cholesterol_Count,
   Total_Count,
   (High_Cholesterol_Count * 100.0 / Total_Count) AS Percentage_High_Cholesterol
FROM
   Family_History_Stats
ORDER BY
   Family_History_Status;
```

Result (raw)

	FAMILY_HISTORY_STATUS	HIGH_CHOLESTEROL_CO...	TOTAL_COUNT	PERCENTAGE_HIGH_CHOLI
1	With Family History	2454	4320	56.805555555555555555555555
2	Without Family History	2600	4443	58.519018681071348188161152

Comments	The outputs in both cases appear to be the same

Observation: The reason we got the same output for our SQL questions for both the normalized tables and the raw table could be either our dataset doesn't contain duplicate data (it's a small dataset with 8763 rows) or the proposed questions and the solutions to the proposed questions to analyze our dataset doesn't contain duplicate data to differ between the results of the normalized table and the raw table.