

# Working with MIMIC Dataset using Google BigQuery

## Query 1: Patient Numbers

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes the Google Cloud logo, a dropdown menu for 'Lab08', a search bar, and links for 'Products, resources, docs (/)'. Below the navigation bar, there's a 'SANDBOX' banner with a link to 'Set up billing to upgrade to the full BigQuery experience. Learn more'. The main interface is divided into three sections: Explorer, Query Editor, and Query Results.

**Explorer:** Shows a search bar and a list of resources. The 'hazel-tome-368000' resource is selected.

**Query Editor:** Displays the SQL query for Query 1:

```
1 SELECT *
2 FROM `physionet-data.mimiciii.clinical.patients`
```

**Query Results:** Shows the results of the query. The table has 11 rows and 8 columns: ROW\_ID, SUBJECT\_ID, GENDER, DOB, DOD, DOB\_HOSP, and DOD\_SSN. The results are displayed in a table format with a 'PREVIEW' tab selected.

Row	ROW_ID	SUBJECT_ID	GENDER	DOB	DOD	DOB_HOSP	DOD_SSN
1	234	249	F	2075-03-13T00:00:00	null	null	null
2	238	253	F	2089-11-26T00:00:00	null	null	null
3	242	258	F	2124-09-19T00:00:00	null	null	null
4	243	260	F	2105-03-23T00:00:00	null	null	null
5	247	264	F	2162-11-30T00:00:00	null	null	null
6	249	266	F	2090-12-17T00:00:00	null	null	null
7	250	267	F	2131-09-05T00:00:00	null	null	null
8	626	663	F	2086-05-12T00:00:00	null	null	null
9	630	667	F	2053-09-17T00:00:00	null	null	null
10	638	675	F	2158-04-05T00:00:00	null	null	null
11	641	678	F	2101-07-09T00:00:00	null	null	null

Results per page: 50 | 1 - 50 of 46520 | < > | REFRESH

## Query 2: Obtain the number of patients with the following query

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar is the same as in the previous screenshot. The main interface is divided into three sections: Explorer, Query Editor, and Query Results.

**Explorer:** Shows the same 'hazel-tome-368000' resource selected.

**Query Editor:** Displays the SQL query for Query 2:

```
1 SELECT COUNT(*)
2 FROM `physionet-data.mimiciii.clinical.patients`;
```

**Query Results:** Shows the results of the query. The table has 1 row and 1 column: COUNT\_1. The result is 46520.

Row	COUNT_1
1	46520

Results per page: 50 | 1 - 50 of 46520 | < > | REFRESH

Query 3: The 'gender' column identifies the gender of the patient. We can obtain the distinct values used to indicate gender using the following query:

The screenshot shows the Google Cloud BigQuery interface. The query editor contains the following SQL code:

```
1 SELECT DISTINCT(gender)
2 FROM `physionet-data.mimiciii.clinical_patients`;
```

The query results are displayed in a table with the following data:

Row	gender
1	F
2	M

The interface also shows the Explorer on the left, the Query results tab selected, and the 'Query completed' status.

Query 4: We can see that 'M' and 'F' are the two characters used to indicate patient gender. We can use this information to obtain the number of female patients by adding a condition to select rows where the gender is 'F':

The screenshot shows the Google Cloud BigQuery interface. The query editor contains the following SQL code:

```
1 SELECT COUNT(*)
2 FROM `physionet-data.mimiciii.clinical_patients`
3 WHERE gender = 'F';
```

The query results are displayed in a table with the following data:

Row	count(*)
1	20399

The interface also shows the Explorer on the left, the Query results tab selected, and the 'Query completed' status.

Query 5: the numbers of male and female patients can be obtained using the following query

The screenshot shows the Google Cloud BigQuery Explorer interface. The query editor contains the following SQL code:

```
1 SELECT gender, COUNT(*)
2 FROM `physionet-data.mimiciii-clinical.patients`
3 GROUP BY gender;
```

The query results are displayed in a table with two columns: gender and fo\_ (representing count). The results are as follows:

Row	gender	fo_
1	F	20399
2	M	26121

Query 6: **Mortality and admissions-** A flag which records whether or not a patient died in the hospital is stored in the patients table. Count the number of patients who died using the following query:

The screenshot shows the Google Cloud BigQuery Explorer interface. The query editor contains the following SQL code:

```
1 SELECT expire_flag, COUNT(*)
2 FROM `physionet-data.mimiciii-clinical.patients`
3 GROUP BY expire_flag;
```

The query results are displayed in a table with two columns: expire\_flag and fo\_ (representing count). The results are as follows:

Row	expire_flag	fo_
1	0	30761
2	1	15759

## Query 7: Patient age and mortality

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes the Google Cloud logo, a dropdown menu for 'Lab08', a search bar, and links for 'Products, resources, docs (/)'. Below the navigation bar, there's a 'Sandbox' section with a message 'Set up billing to upgrade to the full BigQuery experience. Learn more' and buttons for 'DISMISS' and 'UPGRADE'. The main area is titled 'Explorer' and contains a search bar, a list of resources, and a query editor. The query editor shows a SQL query for Query 7, which selects patient information and admission details. The query results are displayed in a table with columns: Row, subject\_id, dob, hadm\_id, admittance, and expire\_flag. The results show 9 rows of data. The interface also includes a 'Query results' section with tabs for 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'EXECUTION GRAPH', and 'PREVIEW'. The 'RESULTS' tab is active, showing the table data. The bottom of the interface has a 'PERSONAL HISTORY' and 'PROJECT HISTORY' section, and a 'REFRESH' button.

```
1 SELECT p.subject_id, p.dob, a.hadm_id,
2       a.admittime, p.expire_flag
3 FROM `physionet-data.mimiciii.clinical_admissions` a
4 INNER JOIN `physionet-data.mimiciii.clinical_patients` p
5 ON p.subject_id = a.subject_id;
```

Row	subject_id	dob	hadm_id	admittime	expire_flag
1	249	2075-03-13T00:00:00	116935	2149-12-17T20:41:00	0
2	249	2075-03-13T00:00:00	158975	2156-04-27T15:33:00	0
3	249	2075-03-13T00:00:00	149546	2155-02-03T20:16:00	0
4	253	2089-11-26T00:00:00	176189	2174-01-21T20:58:00	0
5	258	2124-09-19T00:00:00	189406	2124-09-19T03:59:00	0
6	260	2105-03-23T00:00:00	190363	2105-03-23T10:23:00	0
7	264	2162-11-30T00:00:00	136714	2162-11-30T02:19:00	0
8	266	2090-12-17T00:00:00	186251	2168-07-10T08:01:00	0
9	267	2131-09-05T00:00:00	163714	2156-06-04T05:08:00	0

Query 8: Next we find the earliest admission date for each patient. This requires the use of two functions: the 'MIN' function, which obtains the minimum value, and the 'PARTITION BY' function, which determines the groups over which the minimum value is obtained. To determine the earliest admission time for each patient:

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes the Google Cloud logo, a dropdown menu for 'Lab08', a search bar, and links for 'Products, resources, docs (/)'. Below the navigation bar, there's a 'Sandbox' section with a message 'Set up billing to upgrade to the full BigQuery experience. Learn more' and buttons for 'DISMISS' and 'UPGRADE'. The main area is titled 'Explorer' and contains a search bar, a list of resources, and a query editor. The query editor shows a SQL query for Query 8, which selects patient information and admission details, including the earliest admission time. The query results are displayed in a table with columns: Row, subject\_id, dob, hadm\_id, admittance, expire\_flag, and first\_admittime. The results show 8 rows of data. The interface also includes a 'Query results' section with tabs for 'RESULTS', 'JSON', 'EXECUTION DETAILS', 'EXECUTION GRAPH', and 'PREVIEW'. The 'RESULTS' tab is active, showing the table data. The bottom of the interface has a 'PERSONAL HISTORY' and 'PROJECT HISTORY' section, and a 'REFRESH' button.

```
1 SELECT p.subject_id, p.dob, a.hadm_id,
2       a.admittime, p.expire_flag,
3       MIN(a.admittime) OVER (PARTITION BY p.subject_id) AS first_admittime
4 FROM `physionet-data.mimiciii.clinical_admissions` a
5 INNER JOIN `physionet-data.mimiciii.clinical_patients` p
6 ON p.subject_id = a.subject_id
7 ORDER BY a.hadm_id, p.subject_id;
```

Row	subject_id	dob	hadm_id	admittime	expire_flag	first_admittime
1	58526	2082-03-21T00:00:00	100001	2117-09-11T11:46:00	0	2117-09-11T11:46:00
2	54610	2090-05-19T00:00:00	100003	2150-04-17T15:34:00	1	2150-04-17T15:34:00
3	9895	2059-05-07T00:00:00	100006	2108-04-06T15:49:00	1	2108-04-06T15:49:00
4	23018	2071-06-04T00:00:00	100007	2145-03-31T05:33:00	0	2145-03-31T05:33:00
5	533	2101-07-30T00:00:00	100009	2162-05-16T15:56:00	0	2162-05-16T15:56:00
6	55853	2055-06-03T00:00:00	100010	2109-12-10T07:15:00	0	2109-12-10T07:15:00
7	87977	2156-02-27T00:00:00	100011	2177-08-29T04:51:00	0	2177-08-29T04:51:00
8	60039	2109-06-26T00:00:00	100012	2177-03-12T11:48:00	0	2177-03-12T11:48:00

## Query 9: Let's calculate the age of patients at their time of admission:

The screenshot shows the Google Cloud BigQuery console. The query editor contains the following SQL:

```
1 SELECT p.subject_id, p.dob, a.hadm_id,  
2       a.admittime, p.expire_flag,  
3       DATEDIFF(admittime, dob, YEAR) as age  
4 FROM `physionet-data_mimiciii_clinical.admissions` a  
5 INNER JOIN `physionet-data_mimiciii_clinical.patients` p  
6 ON p.subject_id = a.subject_id  
7 ORDER BY p.subject_id, a.hadm_id;
```

The query results table is displayed below the editor:

Row	subject_id	dob	hadm_id	admittime	expire_flag	age
1	2	2138-07-17T00:00:00	163353	2138-07-17T19:04:00	0	0
2	3	2025-04-11T00:00:00	145834	2101-10-20T19:08:00	1	76
3	4	2143-05-12T00:00:00	185777	2191-03-16T00:28:00	0	48
4	5	2103-02-02T00:00:00	178980	2103-02-02T04:31:00	0	0
5	6	2109-06-21T00:00:00	107064	2175-05-30T07:15:00	0	66
6	7	2121-05-23T00:00:00	118037	2121-05-23T15:05:00	0	0
7	8	2117-11-20T00:00:00	159514	2117-11-20T10:22:00	0	0
8	9	2108-01-26T00:00:00	150750	2149-11-09T13:06:00	1	41

Query 10: If we examine the same patient more than once when calculating a statistic such as mortality, then our estimate will contain “repeated measures”. Unless we handle this phenomenon explicitly, our calculation will be biased.

A simple solution is to only examine the first hospitalization for each patient, which we can do with a GROUP BY clause.

The screenshot shows the Google Cloud BigQuery console. The query editor contains the following SQL:

```
1 SELECT  
2   p.subject_id, p.dob, p.gender  
3   , MIN(a.admittime) AS first_admittime  
4   , MIN( DATEDIFF(admittime, dob, YEAR) )  
5   AS first_admit_age  
6 FROM `physionet-data_mimiciii_clinical.patients` p  
7 INNER JOIN `physionet-data_mimiciii_clinical.admissions` a  
8 ON p.subject_id = a.subject_id  
9 GROUP BY p.subject_id, p.dob, p.gender  
10 ORDER BY p.subject_id
```

The query results table is displayed below the editor:

Row	subject_id	dob	gender	first_admittime	first_admit_age
1	2	2138-07-17T00:00:00	M	2138-07-17T19:04:00	0
2	3	2025-04-11T00:00:00	M	2101-10-20T19:08:00	76
3	4	2143-05-12T00:00:00	F	2191-03-16T00:28:00	48
4	5	2103-02-02T00:00:00	M	2103-02-02T04:31:00	0
5	6	2109-06-21T00:00:00	F	2175-05-30T07:15:00	66
6	7	2121-05-23T00:00:00	F	2121-05-23T15:05:00	0

Query 11: Now that we have a set of unique patients with their age, we can group them into sensible categories based upon age and calculate the mortality rate in these categories. Patients with an age  $\geq 15$  years old are adults and the rest are assigned to other categories. Note the use of the **WITH** clause, which allows us to make a temporary view which we can query against in subsequent lines.

Query 11 SQL:

```

1 WITH first_admission_time AS
2 (
3   SELECT
4     p.subject_id, p.dob, p.gender
5     , MIN(a.admittime) AS first_admittime
6     , MIN(DATETIME_DIFF(admittime, dob, YEAR))
7     AS first_admit_age
8   FROM `physionet-data.mimiciii.clinical.patients` p
9   INNER JOIN `physionet-data.mimiciii.clinical.admissions` a
10  ON p.subject_id = a.subject_id
11  GROUP BY p.subject_id, p.dob, p.gender
12  ORDER BY p.subject_id
13 )
14 SELECT
15   subject_id, dob, gender
16   , first_admittime, first_admit_age

```

Row	subject_id	dob	gender	first_admittime	first_admit	age_group
1	2	2138-07-17T00:00:00	M	2138-07-17T19:04:00	0	neonate
2	3	2025-04-11T00:00:00	M	2101-10-20T19:08:00	76	adult
3	4	2143-05-12T00:00:00	F	2191-03-16T00:28:00	48	adult

Query 12: The above query can now be combined with the **WHERE** and **COUNT** functions described earlier to determine the number of adult patients, whether or not they died, and therefore, their mortality rate.

Query 12 SQL:

```

1 WITH first_admission_time AS
2 (
3   SELECT
4     p.subject_id, p.dob, p.gender
5     , MIN(a.admittime) AS first_admittime
6     , MIN(DATETIME_DIFF(admittime, dob, YEAR))
7     AS first_admit_age
8   FROM `physionet-data.mimiciii.clinical.patients` p
9   INNER JOIN `physionet-data.mimiciii.clinical.admissions` a
10  ON p.subject_id = a.subject_id
11  GROUP BY p.subject_id, p.dob, p.gender
12  ORDER BY p.subject_id
13 )
14 SELECT
15   age_group, gender
16   , COUNT(*) AS NumberOP
17  WHERE age_group = 'adult'
18  GROUP BY age_group, gender

```

Row	age_group	gender	NumberOP
1	adult	F	15476
2	neonate	F	3629
3	>89	F	1294
4	adult	M	21179
5	neonate	M	4245
6	>89	M	697



## Query 13: ICU stays

Google Cloud Lab08 Search Products, resources, docs (/)

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS UPGRADE

Explorer

Q Type to search

Viewing all resources. [Show starred resources only.](#)

hazel-tome-368000 ☆

Query results

SAVE RESULTS EXPLORE DATA

Query completed.

```

1 SELECT *
2 FROM `physionet-data.mimiciii.clinical.transfers`;
```

Press Alt+F1 for Accessibility Options.

Row	ROW_ID	SUBJECT_ID	HADM_ID	ICUSTAY_ID	DBSOURCE	EVENTTYPE	PREV_CAREUNIT	CURR_CAREUNIT
1	3831	690	174817	null	null	null	null	null
2	5690	1010	138250	null	null	null	null	null
3	8152	1490	135580	null	null	null	null	null
4	18207	3369	126808	null	null	null	null	null
5	27059	5008	176430	null	null	null	null	null
6	41025	7544	116945	null	null	null	null	null
7	37239	6884	154499	null	null	null	null	null
8	53086	9768	111591	null	null	null	null	null
9	87837	16213	199739	null	null	null	null	null
10	106511	19620	137540	null	null	null	null	null
11	103520	19092	184323	null	null	null	null	null

Results per page: 50 1 - 50 of 261897

PERSONAL HISTORY PROJECT HISTORY REFRESH

Query 14: The transfers table may have multiple entries per patient to provide detail of all movement between various careunits of the hospital. The first entry in the transfers table for a patient who comes into the ICU will have nothing in the 'prev\_careunit' column. Similarly, the last entry for a patient will have nothing in the 'curr\_careunit'. Entries that have nothing in both previous and current careunit columns indicate that the patient has been transferred between non intensive care units. An example query for one patient and result from the transfers table is shown below. Note that columns 'intime', 'outtime', and 'los' have been truncated.

Google Cloud Lab08 Search Products, resources, docs (/)

SANDBOX Set up billing to upgrade to the full BigQuery experience. [Learn more](#) DISMISS UPGRADE

Explorer

Q Type to search

Viewing all resources. [Show starred resources only.](#)

hazel-tome-368000 ☆

Query results

SAVE RESULTS EXPLORE DATA

Query completed.

```

1 SELECT *
2 FROM `physionet-data.mimiciii.clinical.transfers`
3 WHERE HADM_ID = 112213;
```

Press Alt+F1 for Accessibility Options.

Row	ROW_ID	SUBJECT_ID	HADM_ID	ICUSTAY_ID	DBSOURCE	EVENTTYPE	PREV_CAREUNIT	CURR_CAREUNIT
1	54	12	112213	null	carevue	admit	null	null
2	56	12	112213	232669	carevue	transfer	null	SICU
3	59	12	112213	null	carevue	transfer	SICU	null
4	57	12	112213	null	carevue	transfer	SICU	null
5	55	12	112213	null	carevue	transfer	null	null
6	60	12	112213	null	carevue	discharge	null	null
7	58	12	112213	232669	carevue	transfer	null	SICU

PERSONAL HISTORY PROJECT HISTORY REFRESH

## Query 15: Solutions to the problems in section 8

### Solution to step 1

The screenshot shows the Google Cloud BigQuery interface. The top bar includes the Google Cloud logo, a search bar, and navigation icons. The left sidebar shows the Explorer view with a search bar and a list of resources. The main area displays the query editor with the following SQL query:

```
1 SELECT ie.subject_id, ie.hadm_id, ie.icustay_id,
2       ie.intime, ie.outtime
3 FROM `physionet-data.mimiciii_clinical.icustays` ie;
```

The query results are displayed in a table with the following columns: Row, subject\_id, hadm\_id, icustay\_id, intime, and outtime. The results show 10 rows of data.

Row	subject_id	hadm_id	icustay_id	intime	outtime
1	275	129886	219649	2170-10-07T11:28:53	2170-10-14T14:38:07
2	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11
3	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
4	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
5	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
6	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
7	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
8	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
9	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39
10	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39

The interface also shows options to save results, explore data, and refresh the results.

## Query 16: Solution to step 2

The screenshot shows the Google Cloud BigQuery interface. The top bar includes the Google Cloud logo, a search bar, and navigation icons. The left sidebar shows the Explorer view with a search bar and a list of resources. The main area displays the query editor with the following SQL query:

```
1 SELECT ie.subject_id, ie.hadm_id, ie.icustay_id,
2       ie.intime, ie.outtime,
3       DATETIME_DIFF(adm.admittime, dob, YEAR) AS age
4 -- we use 'ie' as an alias for the icustays table
5 FROM `physionet-data.mimiciii_clinical.icustays` ie
6 -- we use 'pat' as an alias for the patients table
7 INNER JOIN `physionet-data.mimiciii_clinical.patients` pat
8 -- since subject_id is unique for every row in patients, we will get
9 -- one row for every row in icustays (ie)
10 ON ie.subject_id = pat.subject_id
11 INNER JOIN `physionet-data.mimiciii_clinical.admissions` adm ON adm.SUBJECT_ID = pat.SUBJECT_ID;
```

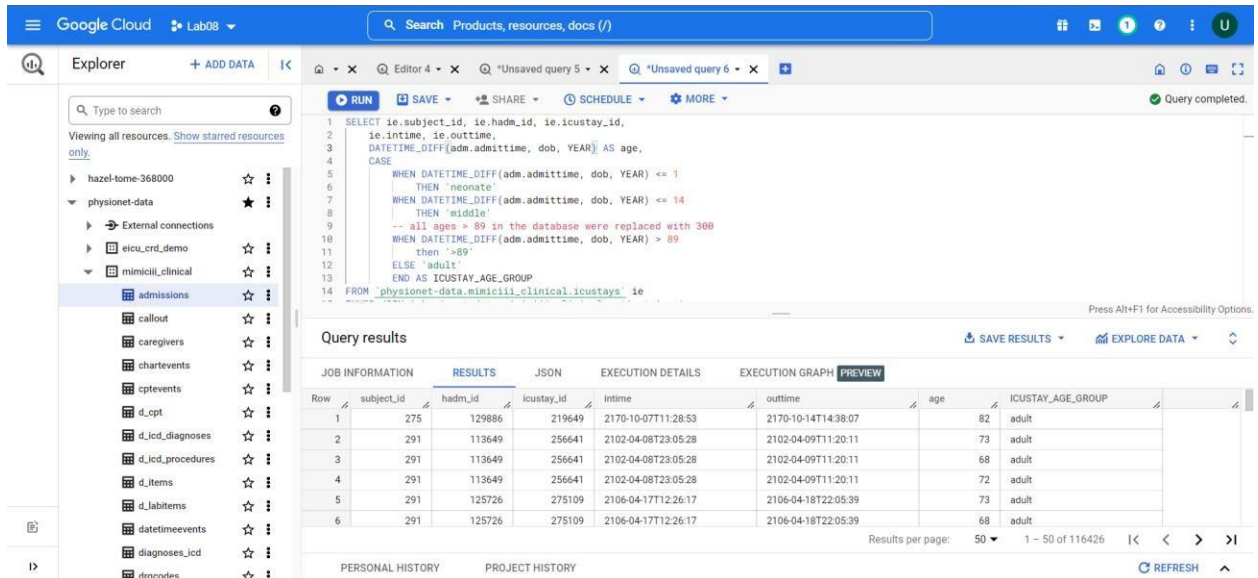
The query results are displayed in a table with the following columns: Row, subject\_id, hadm\_id, icustay\_id, intime, outtime, and age. The results show 10 rows of data.

Row	subject_id	hadm_id	icustay_id	intime	outtime	age
1	275	129886	219649	2170-10-07T11:28:53	2170-10-14T14:38:07	82
2	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11	73
3	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11	68
4	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11	72
5	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	73
6	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	68
7	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	72
8	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	72
9	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	72
10	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	72

The interface also shows options to save results, explore data, and refresh the results.



## Query 17: Solution to step 3



Query 17: Solution to step 3

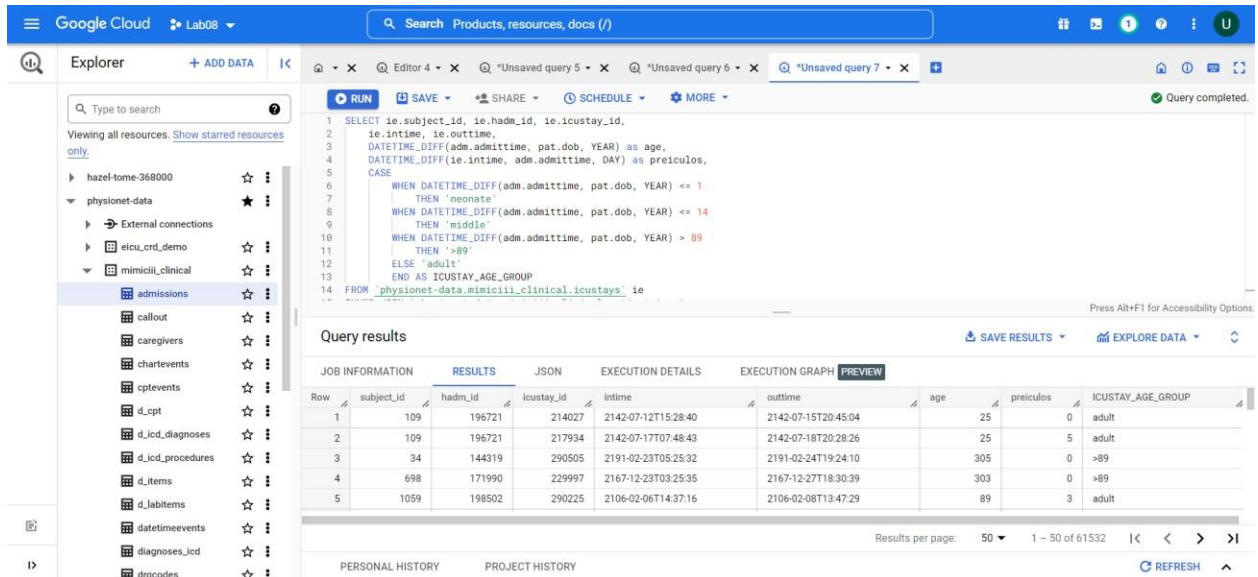
Query SQL:

```
1 SELECT ie.subject_id, ie.hadm_id, ie.icustay_id,
2       ie.intime, ie.outtime,
3       DATETIME_DIFF(adm.admittime, dob, YEAR) AS age,
4       CASE
5         WHEN DATETIME_DIFF(adm.admittime, dob, YEAR) <= 1
6         THEN 'neonate'
7         WHEN DATETIME_DIFF(adm.admittime, dob, YEAR) <= 14
8         THEN 'middle'
9         -- all ages > 89 in the database were replaced with 300
10        WHEN DATETIME_DIFF(adm.admittime, dob, YEAR) > 89
11        THEN '>89'
12        ELSE 'adult'
13        END AS ICUSTAY_AGE_GROUP
14 FROM `physionet-data.mimiciii_clinical.icustays` ie
```

Query results:

Row	subject_id	hadm_id	icustay_id	intime	outtime	age	ICUSTAY_AGE_GROUP
1	275	129886	219649	2170-10-07T11:28:53	2170-10-14T14:38:07	82	adult
2	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11	73	adult
3	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11	68	adult
4	291	113649	256641	2102-04-08T23:05:28	2102-04-09T11:20:11	72	adult
5	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	73	adult
6	291	125726	275109	2106-04-17T12:26:17	2106-04-18T22:05:39	68	adult

## Query 18: Solution to step 4



Query 18: Solution to step 4

Query SQL:

```
1 SELECT ie.subject_id, ie.hadm_id, ie.icustay_id,
2       ie.intime, ie.outtime,
3       DATETIME_DIFF(adm.admittime, pat.dob, YEAR) as age,
4       DATETIME_DIFF(ie.intime, adm.admittime, DAY) as preiculos,
5       CASE
6         WHEN DATETIME_DIFF(adm.admittime, pat.dob, YEAR) <= 1
7         THEN 'neonate'
8         WHEN DATETIME_DIFF(adm.admittime, pat.dob, YEAR) <= 14
9         THEN 'middle'
10        WHEN DATETIME_DIFF(adm.admittime, pat.dob, YEAR) > 89
11        THEN '>89'
12        ELSE 'adult'
13        END AS ICUSTAY_AGE_GROUP
14 FROM `physionet-data.mimiciii_clinical.icustays` ie
```

Query results:

Row	subject_id	hadm_id	icustay_id	intime	outtime	age	preiculos	ICUSTAY_AGE_GROUP
1	109	196721	214027	2142-07-12T15:28:40	2142-07-15T20:45:04	25	0	adult
2	109	196721	217934	2142-07-17T07:48:43	2142-07-18T20:28:26	25	5	adult
3	34	144319	290505	2191-02-23T05:25:32	2191-02-24T19:24:10	305	0	>89
4	698	171990	229997	2167-12-23T03:25:35	2167-12-27T18:30:39	303	0	>89
5	1059	198502	290225	2106-02-06T14:37:16	2106-02-08T13:47:29	89	3	adult

## Query 19: Solution to step 5

The screenshot shows the Google Cloud BigQuery interface. On the left is the Explorer pane with a search bar and a list of resources including 'admissions'. The main editor displays a SQL query that selects patient information and calculates age and preterm status. The query results are shown in a table with columns: subject\_id, hadm\_id, icustay\_id, intime, outtime, deathtime, age, and preiculus. The results table contains 5 rows of data.

```
1 SELECT ie.subject_id, ie.hadm_id, ie.icustay_id,
2       ie.intime, ie.outtime,
3       -- patient death in hospital is stored in the admissions table
4       adm.deathtime,
5       DATETIME_DIFF(adm.admittime, pat.dob, YEAR) as age,
6       DATETIME_DIFF(ie.intime, adm.admittime, DAY) as preiculus,
7       CASE
8         WHEN DATETIME_DIFF(adm.admittime, pat.dob, YEAR) <= 1
9           THEN 'neonate'
10        WHEN DATETIME_DIFF(adm.admittime, pat.dob, YEAR) <= 14
11           THEN 'middle'
12        WHEN DATETIME_DIFF(adm.admittime, pat.dob, YEAR) > 89
13           THEN '>89'
14        ELSE 'adult'
15      END as preterm
16 FROM icustay_data
```

Row	subject_id	hadm_id	icustay_id	intime	outtime	deathtime	age	preiculus
1	109	196721	214027	2142-07-12T15:28:40	2142-07-15T20:45:04	null	25	0
2	109	196721	217934	2142-07-17T07:48:43	2142-07-18T20:28:26	null	25	5
3	34	144319	290505	2191-02-23T05:25:32	2191-02-24T19:24:10	null	305	0
4	698	171990	229997	2167-12-23T03:25:35	2167-12-27T18:30:39	null	303	0
5	1059	198502	290225	2106-02-06T14:37:16	2106-02-08T13:47:29	2106-02-10T07:00:00	89	3

## Query 20: Solution to step 6

The screenshot shows the Google Cloud BigQuery interface with the same SQL query as Query 19. The results table is identical to the one in Query 19, showing 5 rows of patient data with their respective stay details and calculated age/preterm status.

Row	subject_id	hadm_id	icustay_id	intime	outtime	deathtime	age	preiculus
1	109	196721	214027	2142-07-12T15:28:40	2142-07-15T20:45:04	null	25	0
2	109	196721	217934	2142-07-17T07:48:43	2142-07-18T20:28:26	null	25	5
3	34	144319	290505	2191-02-23T05:25:32	2191-02-24T19:24:10	null	305	0
4	698	171990	229997	2167-12-23T03:25:35	2167-12-27T18:30:39	null	303	0
5	1059	198502	290225	2106-02-06T14:37:16	2106-02-08T13:47:29	2106-02-10T07:00:00	89	3

Query 21: Solution to step 7

Google Cloud

Lab08

Search Products, resources, docs (/)

Unsaved query 5

Unsaved query 6

Unsaved query 7

Unsaved query 8

Unsaved query 9

Unsaved query1

Explorer

+ ADD DATA

⌵

Type to search

?

Viewing all resources. Show starred resources only.

hazel-tome-368000

☆

:

physionet-data

★

:

External connections

☆

:

eicu\_crd\_demo

☆

:

mimiclii\_clinical

☆

:

admissions

☆

:

callout

☆

:

caregivers

☆

:

chartevents

☆

:

cptevents

☆

:

d\_cpt

☆

:

d\_icd\_diagnoses

☆

:

d\_icd\_procedures

☆

:

d\_items

☆

:

d\_labitems

☆

:

datetimeevents

☆

:

diagnoses\_icd

☆

:

drgcodes

☆

:

RUN

SAVE

SHARE

SCHEDULE

MORE

Query completed.

6

7

8

9

10

11

12

13

14

15

16

17

DATE\_TIME\_DIFF(ie.intime, adm.admittime, DAY) as preiculos,

CASE

WHEN DATE\_TIME\_DIFF(adm.admittime, pat.dob, YEAR) <= 1

THEN 'neonate'

WHEN DATE\_TIME\_DIFF(adm.admittime, pat.dob, YEAR) <= 14

THEN 'middle'

WHEN DATE\_TIME\_DIFF(adm.admittime, pat.dob, YEAR) > 89

THEN '>89'

ELSE 'adult'

END AS ICUSTAY\_AGE\_GROUP,

-- the "hospital\_expire\_flag" field in the admissions table indicates if a patient died in-hospital

CASE

Press Alt+F1 for Accessibility Options

Query results

SAVE RESULTS

EXPLORE DATA

↕

JOB INFORMATION

RESULTS

JSON

EXECUTION DETAILS

EXECUTION GRAPH

PREVIEW

Row

subject\_id

hadm\_id

icustay\_id

intime

outtime

deathtime

age

preiculos

1

275

129886

219649

2170-10-07T11:28:53

2170-10-14T14:38:07

2170-10-19T15:35:00

82

1

2

291

113649

256641

2102-04-08T23:05:28

2102-04-09T11:20:11

null

68

0

3

291

125726

275109

2106-04-17T12:26:17

2106-04-18T22:05:39

null

72

0

4

294

152578

222074

2118-01-17T21:45:05

2118-01-20T11:12:45

null

79

0

5

301

160332

288401

2189-11-11T12:12:33

2189-11-13T22:11:28

null

85

1

6

304

177469

295659

2141-05-18T17:22:10

2141-05-19T05:40:47

2141-05-19T01:45:00

300

0

Results per page: 50

1 - 50 of 61532

⏪

⏩

⏴

⏵

PERSONAL HISTORY

PROJECT HISTORY

REFRESH

⏶