```
create schema project;

use project;

SELECT

   *

FROM

   hos;

SELECT

   *

FROM

   med;
```

**# 1. To gain a comprehensive understanding of the factors influencing hospitalization costs**

**# a. Merge the two tables by first identifying the columns in the data tables that will help you in merging.**

```
CREATE VIEW hos_med AS

   SELECT

      *

   FROM

      hos

         JOIN

      med USING (`customer id`);

SELECT

   *

FROM

   hos_med;
```

**#  b. In both tables, add a Primary Key constraint for these columns.**

```
alter table hos

modify  `Customer ID` varchar(10) not null;

DELETE FROM hos

WHERE
```

```
   `Customer ID` = '?'; # deleting the rows contains customer id with ?.
SET SQL_SAFE_UPDATES = 0;
alter table hos
add primary key ( `Customer ID`);
alter table med
modify  `Customer ID` varchar(10) not null;
alter table med
add primary key ( `Customer ID`);
SET SQL_SAFE_UPDATES = 1;
```

**# 2. Retrieve information about people who are diabetic and have heart problems with their average age, the average number of dependent children, average BMI, and average hospitalization costs**

**# Method 1**

```
SELECT
   (SELECT
         CASE
            WHEN m.HBA1C > 6.5 THEN 'Yes'
            ELSE 'No'
          END
      ) AS diabetes,
   m.`Heart Issues`,
   ROUND(AVG(2024 - h.year), 0) AS avg_age,
   ROUND(AVG(h.children), 0) AS avg_dep_children,
   ROUND(AVG(m.bmi), 2) AS avg_bmi,
   ROUND(AVG(h.charges), 2) AS avg_hos_costs
FROM
   hos h
      JOIN
   med m USING (`Customer ID`)
```

```
GROUP BY diabetes , m.`heart issues`;
```

**# Method 2**

```
SELECT
   (SELECT
        CASE
            WHEN HBA1C > 6.5 THEN 'Yes'
            ELSE 'No'
          END
      ) AS diabetes,
   `Heart Issues`,
   ROUND(AVG(2024 - year), 0) AS avg_age,
   ROUND(AVG(children), 0) AS avg_dep_children,
   ROUND(AVG(bmi), 2) AS avg_bmi,
   ROUND(AVG(charges), 2) AS avg_hos_costs
FROM
   hos_med
GROUP BY diabetes , `heart issues`;
```

**# 3. Find the average hospitalization cost for each hospital tier and each city level**

**# Method 1**

```
SET SQL_SAFE_UPDATES = 0;
UPDATE hos
SET
   `Hospital tier` = 'tier - 2'
WHERE
   `Hospital tier` = '?';UPDATE hos
SET
   `City tier` = 'tier - 2'
WHERE
   `City tier` = '?';
```

```
SET SQL_SAFE_UPDATES = 1;

SELECT

   `hospital tier`, `city tier`, AVG(charges) AS charges

FROM

   hos

GROUP BY `hospital tier` , `city tier`;
```

**# Method 2**

```
SET SQL_SAFE_UPDATES = 0;

UPDATE hos_med

SET

   `Hospital tier` = 'tier - 2'

WHERE

   `Hospital tier` = '?';

UPDATE hos

SET

   `City tier` = 'tier - 2'

WHERE

   `City tier` = '?';

SET SQL_SAFE_UPDATES = 1;

SELECT

   `hospital tier`, `city tier`, AVG(charges) AS charges

FROM

   hos_med

GROUP BY `hospital tier` , `city tier`;
```

**# 4. Determine the number of people who have had major surgery with a history of cancer.**

**# Method 1**

```
SELECT

   COUNT(*) AS num_pat
```

```
FROM
    (SELECT
        `cancer history`,
            CASE
                WHEN NumberOfMajorSurgeries >= 1 THEN 'Yes'
                ELSE 'No'
            END AS major_surgery
    FROM
        med) m
WHERE
    m.`cancer history` = 'Yes'
        AND m.major_surgery = 'Yes';
```

**# Method 2**

```
SELECT
    COUNT(*) AS num_pat
FROM
    (SELECT
        `cancer history`,
            CASE
                WHEN NumberOfMajorSurgeries >= 1 THEN 'Yes'
                ELSE 'No'
            END AS major_surgery
    FROM
        hos_med) m
WHERE
    `cancer history` = 'Yes'
        AND major_surgery = 'Yes';
```

**# 5.Determine the number of tier-1 hospitals in each state.**

```
SET SQL_SAFE_UPDATES = 0;

DELETE FROM hos

WHERE

    `state ID` = '?';

SET SQL_SAFE_UPDATES = 1;

SELECT

    `State ID`, `Hospital tier`, COUNT(*) AS num_hos

FROM

    hos

GROUP BY `State ID` , `Hospital tier`

HAVING `Hospital tier` = 'tier - 1';
```