

# Vehicle Insurance Database Queries

1. Retrieve Customer and Vehicle details who has been involved in an incident and claim status is pending

## Query :

```
SELECT * FROM T16_CUSTOMER C, T16_VEHICLE V WHERE V.T16_CUST_ID = C.T16_CUST_ID AND C.T16_CUST_ID IN ( SELECT C1.T16_CUST_ID FROM T16_CLAIM C1 WHERE C1.T16_CLAIM_STATUS= 'PENDING' AND C1.T16_INCIDENT_ID IN (SELECT I.T16_INCIDENT_ID FROM T16_INCIDENT_REPORT I WHERE I.T16_CUST_ID =C1.T16_CUST_ID));
```

## Output :

The screenshot displays the MySQL Workbench interface. The SQL Editor contains the following query:

```
-- Query 1
SELECT
*
FROM
T16_CUSTOMER C,
T16_VEHICLE V
WHERE
```

The Results window shows a table with 10 columns: T16\_CUST\_ID, T16\_CUST\_FNAME, T16\_CUST\_UNAME, T16\_CUST\_DOB, T16\_CUST\_GENDER, T16\_CUST\_ADDRESS, T16\_CUST\_MOB\_NUMBER, T16\_CUST\_EMAIL, T16\_CUST\_PASSPORT\_NUMBER, and T16\_CUST\_MARITAL\_S. The data is as follows:

T16_CUST_ID	T16_CUST_FNAME	T16_CUST_UNAME	T16_CUST_DOB	T16_CUST_GENDER	T16_CUST_ADDRESS	T16_CUST_MOB_NUMBER	T16_CUST_EMAIL	T16_CUST_PASSPORT_NUMBER	T16_CUST_MARITAL_S
400	T16_VIJAY	MALLAYA	1992-08-21	M	2/129,3RD_PHASE,GOA	9876543981	VV_MAL@GMAIL.COM	9876-1234-4567	MARRIED
402	T16_SRI	VALLI	1993-12-07	F	AKSHAYA_COLONY,HUBLI	9876543232	VALLI@GMAIL.COM	9876-4444-4567	MARRIED
403	T16_SAI	RAMA	1997-06-15	M	ANNA_COLONY,CHENNAI	9876542332	RAMA@GMAIL.COM	3496-4444-4567	MARRIED
404	T16_KALYAN	BABU	1979-01-18	M	BECENT_ROAD,VIJAYWAD	9381542332	BABU@GMAIL.COM	9124-4444-4567	MARRIED
405	T16_RAM	CHARAN	1985-05-24	M	BANJARA_HILLS,HYD	9381549898	ALWAYS@GMAIL.COM	9124-7342-4567	MARRIED
407	T16_JOE	BIDEN	1944-11-07	M	WHITE_HOUSE,USA	9991543999	JOE@GMAIL.COM	4562-7342-8971	MARRIED
410	T16_ASHLEY	VANCE	1985-06-24	M	KULURU,KURNOOL	7689543789	VANCE@GMAIL.COM	9148-9833-8971	UNMARRIED

The Action Output window shows the following messages:

#	Time	Action	Message	Duration / Fetch
398	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(132.52,"2011-09-26",5700,4CVR13,32.412)	1 row(s) affected	0.000 sec
399	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(133.53,"2018-06-07",5800,4CVR14,33.413)	1 row(s) affected	0.000 sec
400	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(134.54,"2019-05-19",5790,4CVR15,34.414)	1 row(s) affected	0.000 sec
401	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(135.55,"2020-10-24",5800,4CVR16,35.415)	1 row(s) affected	0.000 sec
402	19:01:01	SHOW TABLES	22 row(s) returned	0.015 sec / 0.000 sec
403	19:01:27	SELECT * FROM T16_CUSTOMER C, T16_VEHICLE V WHERE V.T16_CUST_ID = C.T16_CUST...	16 row(s) returned	0.016 sec / 0.000 sec

Query Completed

2. Retrieve customer details who has premium payment amount greater than the sum of all the customerIds in the database

### Query :

```
SELECT * FROM T16_CUSTOMER C WHERE C.T16_CUST_ID IN (SELECT  
P.T16_CUST_ID FROM T16_PREMIUM_PAYMENT P WHERE P.T16_CUST_ID =  
C.T16_CUST_ID AND P.T16_PREMIUM_PAYMENT_AMOUNT > ( SELECT  
SUM(T16_CUST_ID) FROM T16_CUSTOMER ));
```

### Output :

The screenshot displays the MySQL Workbench interface. The SQL Editor at the top contains a query that filters customers based on their premium payment amount relative to the sum of all customer IDs. The Results pane below shows a table with 10 columns: T16\_CUST\_ID, T16\_CUST\_FNAME, T16\_CUST\_LNAME, T16\_CUST\_DOB, T16\_CUST\_GENDER, T16\_CUST\_ADDRESS, T16\_CUST\_MOB\_NUMBER, T16\_CUST\_EMAIL, T16\_CUST\_PASSPORT\_NUMBER, and T16\_CUST\_MARITAL\_S. The table lists 11 customer records. The bottom pane shows the 'Action Output' log, detailing the execution of various SQL commands including INSERT, SHOW TABLES, and the final SELECT query, along with the number of rows affected or returned.

T16_CUST_ID	T16_CUST_FNAME	T16_CUST_LNAME	T16_CUST_DOB	T16_CUST_GENDER	T16_CUST_ADDRESS	T16_CUST_MOB_NUMBER	T16_CUST_EMAIL	T16_CUST_PASSPORT_NUMBER	T16_CUST_MARITAL_S
400	T16_VIJAY	MALLAYA	1992-08-21	M	2/129,3RD_PHASE,GOA	9876543981	VV_MAL@GMAIL.COM	9876-1234-4567	MARRIED
402	T16_SRI	VALLI	1993-12-07	F	AKSHAYA_COLONY,HUBLI	9876543232	VALLI@GMAIL.COM	9876-4444-4567	MARRIED
403	T16_SAI	RAMA	1997-06-15	M	ANNA_COLONY,CHENNAI	9876542332	RAMA@GMAIL.COM	3496-4444-4567	MARRIED
405	T16_RAM	CHARAN	1985-05-24	M	BANJARA_HILLS,HYD	9381549898	ALWAYS@GMAIL.COM	9124-7342-4567	MARRIED
406	T16_PS	KARTHICK	1964-02-28	M	ADYAR,CHENNAI	9381543625	PSK@GMAIL.COM	4562-7342-4567	UNMARRIED
408	T16_ELON	MUSK	1976-03-14	M	SEA_ATTLE,CALIFORNIA	9991543789	MUSK@GMAIL.COM	1594-7342-8971	MARRIED
409	T16_JACK	MA	1971-07-19	M	ALI_BABA,TAIWAN	8011543789	JACK@GMAIL.COM	4951-7342-8971	MARRIED
410	T16_ASHLEY	VANCE	1985-06-24	M	KULLURU,KURNOOL	7689543789	VANCE@GMAIL.COM	9148-9833-8971	UNMARRIED
411	T16_AYN	RAND	1956-09-11	F	AMSTER,NETHERLANDS	6362297167	RAND@GMAIL.COM	7167-2363-8971	UNMARRIED

3. Retrieve Company details whose number of products is greater than departments, where the departments are located in more than one location

### Query :

```
SELECT C.T16_COMPANY_NAME FROM T16_INSURANCE_COMPANY C INNER  
JOIN T16_PRODUCT P ON P.T16_COMPANY_NAME=C.T16_COMPANY_NAME  
GROUP BY P.T16_COMPANY_NAME HAVING COUNT(*) >ALL(SELECT COUNT(*)  
FROM T16_INSURANCE_COMPANY GROUP BY T16_COMPANY_NAME HAVING  
COUNT(T16_COMPANY_LOCATION)>1);
```

### Output :

The screenshot shows the MySQL Workbench interface. The SQL Editor contains a query labeled 'Query 3' which is the same as the one provided in the previous block. The 'Result Grid' shows the output of the query, displaying two rows of company names: 'AEGON INSURANCE' and 'ICICI PRUDENTIAL'. The 'Output' pane at the bottom shows the execution log, including the query execution time and the number of rows returned.

#	Time	Action	Message	Duration / Fetch
400	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(134.54,"2019-05-19",5790,4CVR15,94,414)	1 row(s) affected	0.000 sec
401	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(135.55,"2020-10-24",5800,4CVR16,95,415)	1 row(s) affected	0.000 sec
402	19:01:01	SHOW TABLES	22 row(s) returned	0.015 sec / 0.000 sec
403	19:01:27	SELECT * FROM T16_CUSTOMER C, T16_VEHICLE V WHERE V.T16_CUST_ID = C.T16_CUST...	16 row(s) returned	0.016 sec / 0.000 sec
404	19:01:55	SELECT * FROM T16_CUSTOMER C WHERE C.T16_CUST_ID IN (SELECT P.T16_CUST_...	12 row(s) returned	0.015 sec / 0.000 sec
405	19:02:20	SELECT DISTINCT (T16_COMPANY_NAME) FROM T16_INSURANCE_COMPANY WHERE T16_C...	2 row(s) returned	0.016 sec / 0.000 sec

4. Select Customers who have more than one Vehicle, where the premium for one of the Vehicles is not paid and it is involved in accident

**Query :**

```
SELECT CONCAT(C.T16_CUST_ID,',',T16_CUST_FNAME,T16_CUST_LNAME) AS  
CUST_INFO FROM (((T16_CUSTOMER C INNER JOIN T16_VEHICLE V ON  
(C.T16_CUST_ID=V.T16_CUST_ID) )INNER JOIN T16_PREMIUM_PAYMENT P ON  
(P.T16_CUST_ID=V.T16_CUST_ID AND P.T16_PREMIUM_PAYMENT_AMOUNT=0))  
INNER JOIN T16_INCIDENT_REPORT IR ON (IR.T16_CUST_ID=P.T16_CUST_ID  
AND IR.T16_INCIDENT_TYPE='ACCIDENT')) GROUP BY V.T16_CUST_ID HAVING  
COUNT(V.T16_VEHICLE_ID)>1;
```

The screenshot displays the MySQL Workbench interface. The 'SQL File 3' editor shows the following query:

```
-- Query 4  
889  
890 SELECT  
891     CONCAT(C.T16_CUST_ID,  
892           ',',  
893           T16_CUST_FNAME,  
894           T16_CUST_LNAME) AS CUST_INFO  
895 FROM  
896     (((T16_CUSTOMER C  
897       INNER JOIN T16_VEHICLE V ON (C.T16_CUST_ID = V.T16_CUST_ID))
```

The 'Result Grid' shows the output of the query:

CUST_INFO
411,T16_AYNRAND
413,T16_JAYAPRAKASH
402,T16_SRIYALI
403,T16_SAIRAMA

The 'Output' pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
401	19:01:01	INSERT INTO T16_CLAIM_SETTLEMENT VALUES(135.55,'2020-10-24','5800,4CVR16',95.415)	1 row(s) affected	0.000 sec
402	19:01:01	SHOW TABLES	22 row(s) returned	0.015 sec / 0.000 sec
403	19:01:27	SELECT * FROM T16_CUSTOMER C, T16_VEHICLE V WHERE V.T16_CUST_ID = C.T16_CUST...	16 row(s) returned	0.016 sec / 0.000 sec
404	19:01:55	SELECT * FROM T16_CUSTOMER C WHERE C.T16_CUST_ID IN (SELECT P.T16_CUST...	12 row(s) returned	0.015 sec / 0.000 sec
405	19:02:20	SELECT DISTINCT (T16_COMPANY_NAME) FROM T16_INSURANCE_COMPANY WHERE T16_C...	2 row(s) returned	0.016 sec / 0.000 sec
406	19:02:48	SELECT CONCAT(C.T16_CUST_ID, ',', T16_CUST_FNAME, T16_CUST_LNAME)...	4 row(s) returned	0.016 sec / 0.000 sec

5. Select all vehicles which have premium more than its vehicle number.

**Query :**

```
SELECT distinct(V.T16_VEHICLE_ID) FROM T16_VEHICLE V,  
T16_PREMIUM_PAYMENT P WHERE V.T16_CUST_ID = P.T16_CUST_ID AND  
P.T16_PREMIUM_PAYMENT_ID > V.T16_VEHICLE_NUMBER;
```

**Output :**

The screenshot displays the MySQL Workbench interface. The SQL Editor at the top contains the following query:

```
-- Query 5  
SELECT DISTINCT  
  (V.T16_VEHICLE_ID)  
FROM  
  T16_VEHICLE V,  
  T16_PREMIUM_PAYMENT P  
WHERE  
  V.T16_CUST_ID = P.T16_CUST_ID
```

The Results pane below the editor shows a table with one column, `T16_VEHICLE_ID`, and six rows of data. The values in the rows are 40, 41, 42, 43, 44, and 45.

The Output pane at the bottom shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
402	19:01:01	SHOW TABLES	22 row(s) returned	0.015 sec / 0.000 sec
403	19:01:27	SELECT * FROM T16_CUSTOMER C, T16_VEHICLE V WHERE V.T16_CUST_ID = C.T16_CUST_ID	16 row(s) returned	0.016 sec / 0.000 sec
404	19:01:55	SELECT * FROM T16_CUSTOMER C WHERE C.T16_CUST_ID IN (SELECT P.T16_CUST_ID FROM T16_PREMIUM_PAYMENT P)	12 row(s) returned	0.015 sec / 0.000 sec
405	19:02:20	SELECT DISTINCT (T16_COMPANY_NAME) FROM T16_INSURANCE_COMPANY WHERE T16_C...	2 row(s) returned	0.016 sec / 0.000 sec
406	19:02:48	SELECT CONCAT(C.T16_CUST_ID, ' ', T16_CUST_FNAME, T16_CUST_LNAME) ...	4 row(s) returned	0.016 sec / 0.000 sec
407	19:03:09	SELECT DISTINCT (V.T16_VEHICLE_ID) FROM T16_VEHICLE V, T16_PREMIUM_PAYMENT P W...	19 row(s) returned	0.000 sec / 0.000 sec

6. Retrieve Customer details whose Claim Amount is less than Coverage Amount and Claim Amount is greater than Sum of (CLAIM\_SETTLEMENT\_ID, VEHICLE\_ID, CLAIM\_ID, CUST\_ID )

### Query :

```
SELECT CONCAT(C.T16_CUST_ID,"",T16_CUST_FNAME,"",T16_CUST_LNAME)
AS CUST_INFO FROM (T16_CUSTOMER C INNER JOIN
T16_CLAIM_SETTLEMENT CS ON CS.T16_CUST_ID=C.T16_CUST_ID INNER JOIN
T16_CLAIM CL ON (CL.T16_CLAIM_ID=CS.T16_CLAIM_ID)INNER JOIN
T16_COVERAGE CO ON (CO.T16_COVERAGE_ID=CS.T16_COVERAGE_ID AND
CL.T16_CLAIM_AMOUNT<CO.T16_COVERAGE_AMOUNT AND
CL.T16_CLAIM_AMOUNT>(CS.T16_CLAIM_SETTLEMENT_ID+CS.T16_VEHICLE_ID
+CS.T16_CLAIM_ID+CS.T16_CUST_ID))));
```

### Output :

The screenshot shows the MySQL Workbench interface. The SQL Editor contains the following query:

```
(V.T16_VEHICLE_ID)
-- Execute the selected portion of the script or everything, if there is no selection
T16_VEHICLE V,
T16_PREMIUM_PAYMENT P
WHERE
V.T16_CUST_ID = P.T16_CUST_ID
AND P.T16_PREMIUM_PAYMENT_ID > V.T16_VEHICLE_NUMBER;
```

The Results Grid shows the output of the query, which is a list of customer information:

CUST_INFO
401,T16_ROHIT,KUMAR
402,T16_SRI,VALLI
403,T16_SAI,RAMA
404,T16_KALYAN,BABU
405,T16_RAM,CHARAN
407,T16_JOE,BIDEN
410,T16_ASHLEY,VANCE

The Output tab shows the execution log with the following details:

#	Time	Action	Message	Duration / Fetch
403	19:01:27	SELECT * FROM T16_CUSTOMER C, T16_VEHICLE V WHERE V.T16_CUST_ID = C.T16_CUST...	16 row(s) returned	0.016 sec / 0.000 sec
404	19:01:55	SELECT * FROM T16_CUSTOMER C WHERE C.T16_CUST_ID IN (SELECT P.T16_CUST...	12 row(s) returned	0.015 sec / 0.000 sec
405	19:02:20	SELECT DISTINCT (T16_COMPANY_NAME) FROM T16_INSURANCE_COMPANY WHERE T16_C...	2 row(s) returned	0.016 sec / 0.000 sec
406	19:02:48	SELECT CONCAT(C.T16_CUST_ID, '', T16_CUST_FNAME, T16_CUST_LNAME) ...	4 row(s) returned	0.016 sec / 0.000 sec
407	19:03:09	SELECT DISTINCT (V.T16_VEHICLE_ID) FROM T16_VEHICLE V, T16_PREMIUM_PAYMENT P W...	19 row(s) returned	0.000 sec / 0.000 sec
408	19:03:20	SELECT CONCAT(C.T16_CUST_ID, '', T16_CUST_FNAME, T16_CUST_LNAME) AS CUST_INFO FROM ...	7 row(s) returned	0.016 sec / 0.000 sec