

1. Create database, table and insert data

The screenshot shows the SQL Developer interface. The 'SCHEMAS' pane on the left shows the 'student' schema with a table named 'tblparents'. The 'Query' window contains the following SQL statement:

```
1 • SELECT * FROM student.tblparents;
```

The 'Result Grid' is currently empty, showing only the column headers: parent_id, parent_name, city, age, annual_income, occupation, and email_id. The status bar at the bottom indicates 'Limit to 1000 rows'.

The screenshot shows the same SQL Developer interface, but the 'Result Grid' is now populated with 20 rows of data. The data is as follows:

| parent_id | parent_name | city | age | annual_income | occupation | email_id |
|-----------|-----------------|---------------|-----|---------------|----------------|-------------------|
| 101 | Rajiv sharma | Delhi | 45 | 500000 | Developer | rajiv@gmail.com |
| 102 | Shyam Patel | Panjab | 42 | 830000 | Photographer | shyam@gmail.com |
| 103 | Satyam Kapoor | Ahmedabad | 46 | 550000 | Teacher | satyam@gmail.com |
| 104 | Avdesh kumar | Mumbai | 55 | 600000 | Engineer | avdesh@gmail.com |
| 105 | Ravi Chauhan | chennai | 50 | 650000 | Doctor | ravi@gmail.com |
| 106 | Radhika | Bangalore | 40 | 350000 | Nurse | radhika@gmail.com |
| 107 | Rohit chopra | Bangalore | 35 | 700000 | Engineer | rohit@gmail.com |
| 108 | Ritika Malhotra | Mumbai | 55 | 600000 | Teacher | ritika@gmail.com |
| 109 | Pradeep Singh | Delhi | 45 | 660000 | Musician | pradeep@gmail.com |
| 110 | Ankit Goel | Mohali | 53 | 850000 | Engineer | ankit@gmail.com |
| 111 | Raj kumar | Gujarat | 58 | 560000 | Accountant | raj@gmail.com |
| 112 | Radha kumar | Haryana | 51 | 750000 | Engineer | radha@gmail.com |
| 113 | Deepak kumar | Uttar Prad... | 45 | 520000 | Teacher | deepak@gmail.com |
| 114 | Soni verma | Pune | 49 | 590000 | Doctor | soni@gmail.com |
| 115 | Priya Josi | chennai | 42 | 620000 | Engineer | priya@gmail.com |
| 116 | Kajal kumari | Uttarakhand | 52 | 600000 | Teacher | kajal@gmail.com |
| 117 | Aachal singh | Madhya pr... | 58 | 450000 | Engineer | aachal@gmail.com |
| 118 | Pratap kumar | Bangalore | 55 | 900000 | Software En... | pratap@gmail.com |
| 119 | Lokesh kumar | Ahmedabad | 56 | 700000 | Doctor | lokesh@gmail.com |
| 120 | Varun kumar | Mumbai | 55 | 650000 | Engineer | varun@gmail.com |

The 'SCHEMAS' pane on the left shows the 'tblparents' table structure:

Table: **tblparents**

Columns:

- parent_id int PK
- parent_name varchar
- city varchar
- age int
- annual_income decimal
- occupation varchar
- email_id varchar

The 'Result Grid' is now populated with 20 rows of data. The status bar at the bottom indicates 'Limit to 1000 rows'.

2. Update email id of each parents as NA

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying the following SQL script:

```
1 use student;
2 update tblparents set email_id="NA";
3 select * from tblparents;
4 commit;
```

The 'Result Grid' shows the data from the 'tblparents' table. The columns are: parent_id, parent_name, city, age, annual_income, occupation, and email_id. The data is as follows:

| parent_id | parent_name | city | age | annual_income | occupation | email_id |
|-----------|-----------------|---------------|-----|---------------|----------------|----------|
| 101 | Rajiv sharma | Delhi | 45 | 500000 | Developer | NA |
| 102 | Shyam Patel | Panjab | 42 | 830000 | Photographer | NA |
| 103 | Satyam Kapoor | Ahmedabad | 46 | 550000 | Teacher | NA |
| 104 | Avdesh kumar | Mumbai | 55 | 600000 | Engineer | NA |
| 105 | Ravi Chauhan | chennai | 50 | 650000 | Doctor | NA |
| 106 | Radhika | Bangalore | 40 | 350000 | Nurse | NA |
| 107 | Rohit chopra | Bangalore | 35 | 700000 | Engineer | NA |
| 108 | Ritika Malhotra | Mumbai | 55 | 600000 | Teacher | NA |
| 109 | Pradeep Singh | Delhi | 45 | 660000 | Musician | NA |
| 110 | Ankit Goel | Mohali | 53 | 850000 | Engineer | NA |
| 111 | Raj kumar | Gujarat | 58 | 560000 | Accountant | NA |
| 112 | Radha kumar | Haryana | 51 | 750000 | Engineer | NA |
| 113 | Deepak kumar | Uttar Prad... | 45 | 520000 | Teacher | NA |
| 114 | Soni verma | Pune | 49 | 590000 | Doctor | NA |
| 115 | Priya Josi | chennai | 42 | 620000 | Engineer | NA |
| 116 | Kajal kumari | Uttarakhand | 52 | 600000 | Teacher | NA |
| 117 | Aachal singh | Madhya pr... | 58 | 450000 | Engineer | NA |
| 118 | Pratap kumar | Bangalore | 55 | 900000 | Software En... | NA |
| 119 | Lokesh kumar | Ahmedabad | 56 | 700000 | Doctor | NA |
| 120 | Varun kumar | Mumbai | 55 | 650000 | Engineer | NA |

The 'Information' panel on the left shows the table 'tblparents' with columns: parent_id (int), parent_name (varchar), city (varchar), age (int), annual_income (decimal), occupation (varchar), and email_id (varchar).

3. Count Number of parents whose annual income more than 6 lakh

The screenshot displays the MySQL Workbench interface. The left sidebar shows the 'SCHEMAS' panel with a tree view containing 'student', 'tblparents', 'Columns', 'Indexes', 'Foreign Keys', 'Triggers', 'Views', 'Stored Procedures', 'Functions', and 'sys'. The 'tblparents' table is selected. The main query editor shows the following SQL code:

```
1 use student;
2 select count(Parent_id) from tblparents
3 where annual_Income>=600000;
```

The 'Result Grid' tab is active, showing the query results:

| count(Parent_id) |
|------------------|
| 13 |

The bottom panel shows the 'Information' tab with details for the 'Table: tblparents'.

Table: tblparents

Columns:

| Column Name | Data Type | Nullable |
|---------------|-----------|----------|
| parent_id | int | PI |
| parent_name | varchar | |
| city | varchar | |
| age | int | |
| annual_income | dec | |
| occupation | varchar | |
| email_id | varchar | |

The bottom status bar indicates 'Read Only'.

4. Select parents who are living in “Ahmedabad”, “Delhi”, “Mumbai”, “Chennai”, “Bangalore”.

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying the following SQL query:

```
1 use student;
2 select * from tblparents where city in('Ahmedabad','Delhi','Mumbai','Chennai','Bangalore');
3 commit;
```

The 'Result Grid' shows the following data:

| parent_id | parent_name | city | age | annual_income | occupation | email_id |
|-----------|-----------------|-----------|------|---------------|------------|----------|
| 101 | Rajiv sharma | Delhi | 45 | 500000 | Developer | NA |
| 103 | Satyam Kapoor | Ahmedabad | 46 | 550000 | Teacher | NA |
| 104 | Avdesh kumar | Mumbai | 55 | 600000 | Engineer | NA |
| 105 | Ravi Chauhan | chennai | 50 | 650000 | Doctor | NA |
| 108 | Ritika Malhotra | Mumbai | 55 | 600000 | Teacher | NA |
| 109 | Pradeep Singh | Delhi | 45 | 660000 | Musician | NA |
| 115 | Priya Josi | chennai | 42 | 620000 | Engineer | NA |
| 119 | Lokesh kumar | Ahmedabad | 56 | 700000 | Doctor | NA |
| 120 | Varun kumar | Mumbai | 55 | 650000 | Engineer | NA |
| NULL | NULL | NULL | NULL | NULL | NULL | NULL |

The 'Information' panel on the left shows the table structure for 'tblparents':

Table: **tblparents**

Columns:

- parent_id int(11) PK
- parent_name varchar(50)
- city varchar(50)
- age int(11)
- annual_income decimal(10,2)
- occupation varchar(50)
- email_id varchar(50)

5. Select parents whose annual income is less than 5 lakh otherwise age is more than 45

The screenshot shows the MySQL Workbench interface. The 'Query 1' tab is active, displaying the following SQL query:

```
1 use student;
2 select * from tblparents where annual_income<500000 or age>45;
3 commit;
```

The 'Result Grid' shows the results of the query, displaying 12 rows of data from the 'tblparents' table. The columns are: parent_id, parent_name, city, age, annual_income, occupation, and email_id.

| parent_id | parent_name | city | age | annual_income | occupation | email_id |
|-----------|-----------------|--------------|-----|---------------|--------------|----------|
| 103 | Satyam Kapoor | Ahmedabad | 46 | 550000 | Teacher | NA |
| 104 | Avdesh kumar | Mumbai | 55 | 600000 | Engineer | NA |
| 105 | Ravi Chauhan | chennai | 50 | 650000 | Doctor | NA |
| 106 | Radhika | Bangalore | 40 | 350000 | Nurse | NA |
| 108 | Ritika Malhotra | Mumbai | 55 | 600000 | Teacher | NA |
| 110 | Ankit Goel | Mohali | 53 | 850000 | Engineer | NA |
| 111 | Raj kumar | Gujarat | 58 | 560000 | Accountant | NA |
| 112 | Radha kumar | Haryana | 51 | 750000 | Engineer | NA |
| 114 | Soni verma | Pune | 49 | 590000 | Doctor | NA |
| 116 | Kajal kumari | Uttarakhand | 52 | 600000 | Teacher | NA |
| 117 | Aachal singh | Madhya pr... | 58 | 450000 | Engineer | NA |
| 118 | Pratap kumar | Bangalore | 55 | 900000 | Software ... | NA |
| 119 | Lokesh kumar | Ahmedabad | 56 | 700000 | Doctor | NA |
| 120 | Varun kumar | Mumbai | 55 | 650000 | Engineer | NA |

The 'Table: tblparents' information panel on the left shows the following columns and data types:

- parent_id: int PK
- parent_name: varchar
- city: varchar
- age: int
- annual_income: decimal
- occupation: varchar
- email_id: varchar

6. create table student and insert data

The screenshot displays the MySQL Workbench interface for a local instance of MySQL 8.0. The left sidebar shows the 'SCHEMAS' tree with 'student' expanded, revealing the 'tblparents' table. The 'Information' pane shows the structure of 'tblparents'.

Table: tblparents

Columns:

- parent_id: int PK
- parent_name: varchar(30)
- city: varchar(30)
- age: int
- annual_income: decimal(8,0)
- occupation: varchar(25)
- email_id: varchar(35)

The main query editor shows a query: `SELECT * FROM student.student;` The 'Result Grid' displays the data for the 'student' table.

| student_id | student_name | Maths_marks | science_marks | parent_id |
|------------|--------------|-------------|---------------|-----------|
| 1 | Himani | 90 | 89 | 101 |
| 2 | Diksha | 65 | 72 | 102 |
| 3 | Shivani | 69 | 55 | 101 |
| 4 | Shiv | 45 | 52 | 117 |
| 5 | Deepak | 69 | 75 | 119 |
| 6 | Chandan | 35 | 45 | 104 |
| 7 | Pooja | 78 | 67 | 115 |
| 8 | Neha | 75 | 49 | 113 |
| 9 | Meher | 52 | 74 | 111 |
| 10 | Manisha | 85 | 75 | 106 |
| * | NULL | NULL | NULL | NULL |

The bottom pane shows the 'Output' section with 'student 1' selected.