

This screenshot shows the MySQL Workbench interface with a script editor containing SQL commands to create a database and a table. The 'Schemas' pane on the left shows a tree view of databases, with 'DataScience24' highlighted. The script editor contains the following SQL code:

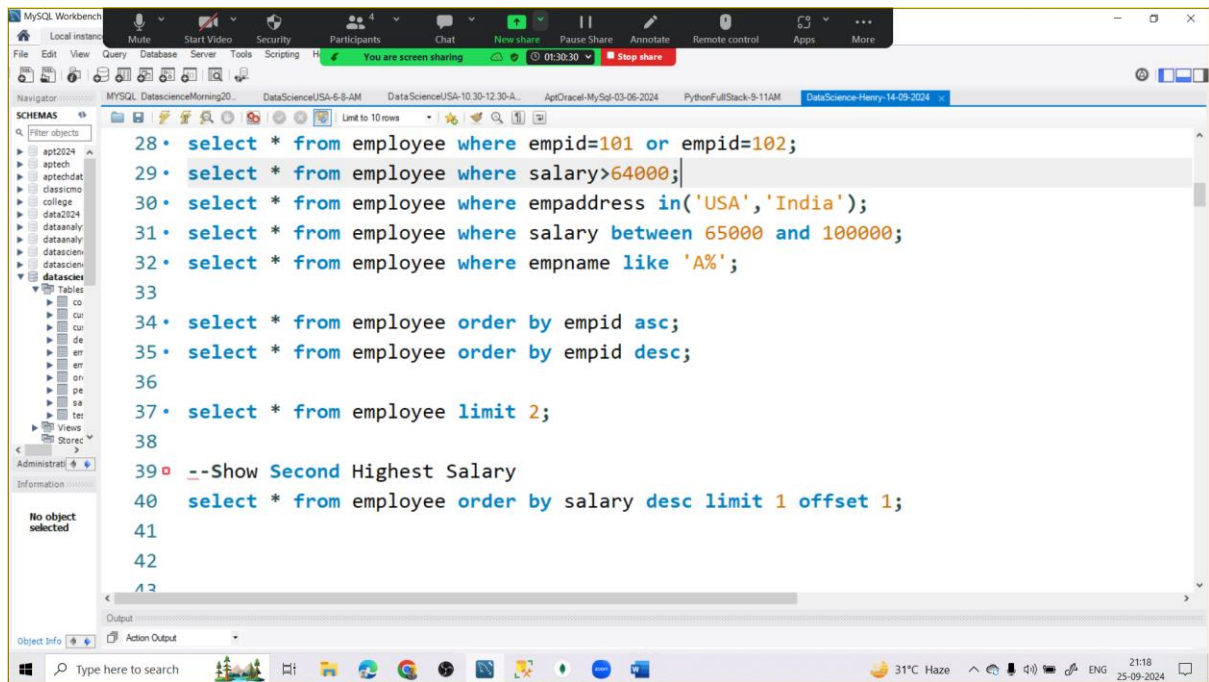
```
1 • create database DataScience24;
2 • use DataScience24;
3
4 • create table employee
5 (
6   empid int primary key not null,
7   empname char(200) not null,
8   empAddress varchar(500),
9   dob date,
10  salary double
11 )
12 select * from employee;
13 • insert into employee values(101,'James','USA','2020-05-06',56225.25);
14 • insert into employee values(102,'Anita','India','2021-06-02',65898),
15                               (103,'Bob','Kanada','2024-04-03',60200);
16 • insert into employee(empid,empname,empaddress) values(104,'Kellon','Taxes');
```

The bottom status bar shows the system time as 21:17 on 25-09-2024.

This screenshot shows the MySQL Workbench interface with a script editor containing SQL commands to filter, update, and delete data from the 'employee' table. The 'Schemas' pane on the left shows a tree view of databases, with 'DataScience24' highlighted. The script editor contains the following SQL code:

```
15                               (103,'Bob','Kanada','2024-04-03',60200);
16 • insert into employee(empid,empname,empaddress) values(104,'Kellon','Taxes');
17
18 • --where use for filtration
19 select * from employee where empid=101;
20
21 • --update for change values
22 update employee set salary=80000 where empid=101;
23 • select * from employee;
24
25 • --Delete values
26 delete from employee where empid=104;
27
28 • select * from employee where empid=101 or empid=102;
29 • select * from employee where salary>64000;
30 • select * from employee where empaddress in('USA','India');
```

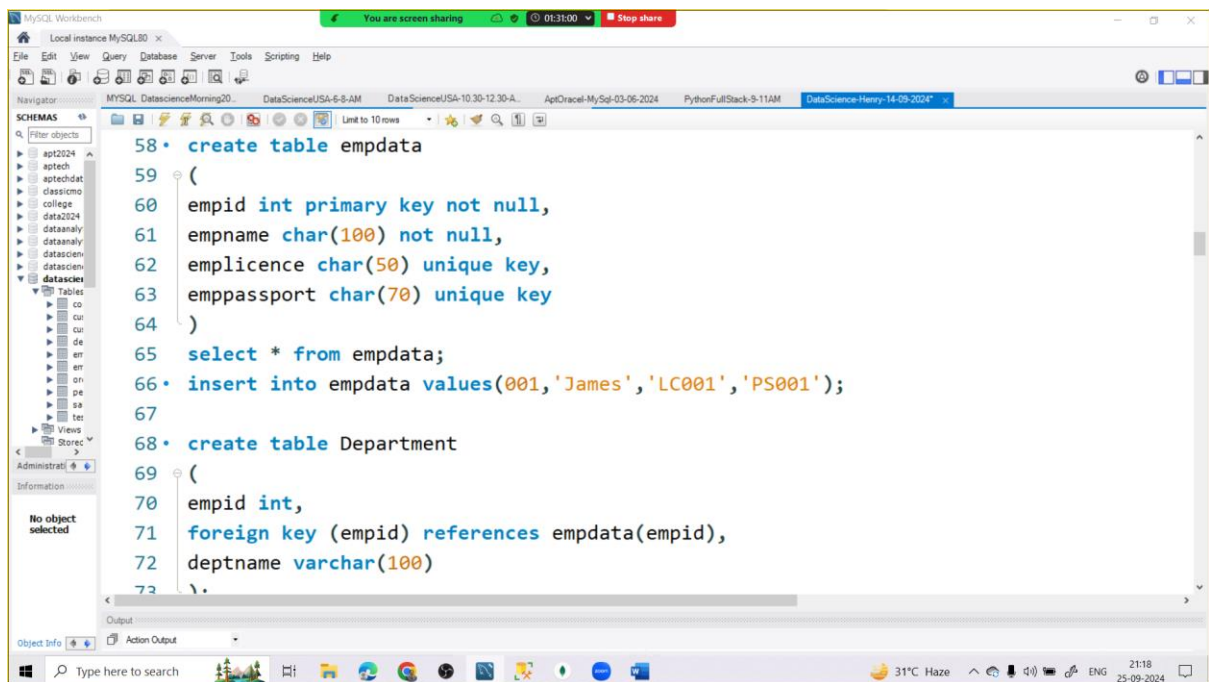
The bottom status bar shows the system time as 21:17 on 25-09-2024.



This screenshot shows the MySQL Workbench interface with a local instance of MySQL. The SQL editor contains a series of queries numbered 28 to 43. The queries are as follows:

```
28 • select * from employee where empid=101 or empid=102;
29 • select * from employee where salary>64000;
30 • select * from employee where empaddress in('USA','India');
31 • select * from employee where salary between 65000 and 100000;
32 • select * from employee where empname like 'A%';
33
34 • select * from employee order by empid asc;
35 • select * from employee order by empid desc;
36
37 • select * from employee limit 2;
38
39 • --Show Second Highest Salary
40 select * from employee order by salary desc limit 1 offset 1;
41
42
43
```

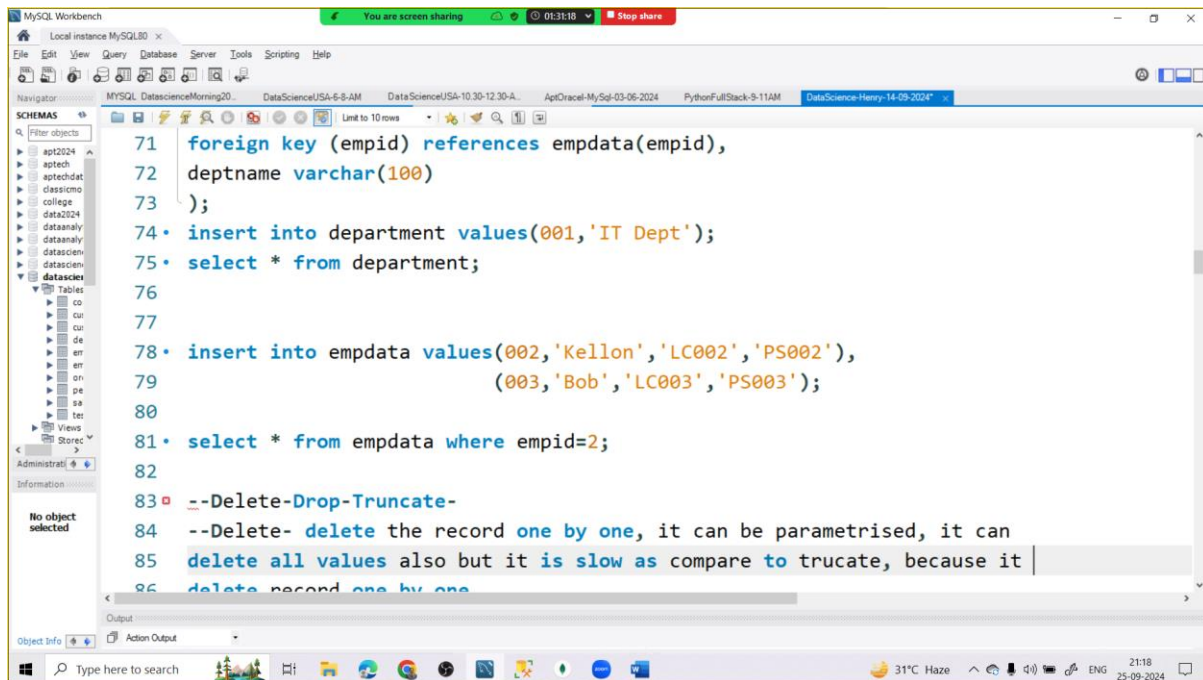
The interface includes a Navigator pane on the left showing a schema with various tables like 'ap12024', 'aptech', 'classico', etc. The bottom status bar shows the system time as 21:18 on 25-09-2024.



This screenshot shows the MySQL Workbench interface with a local instance of MySQL. The SQL editor contains a series of queries numbered 58 to 73. The queries are as follows:

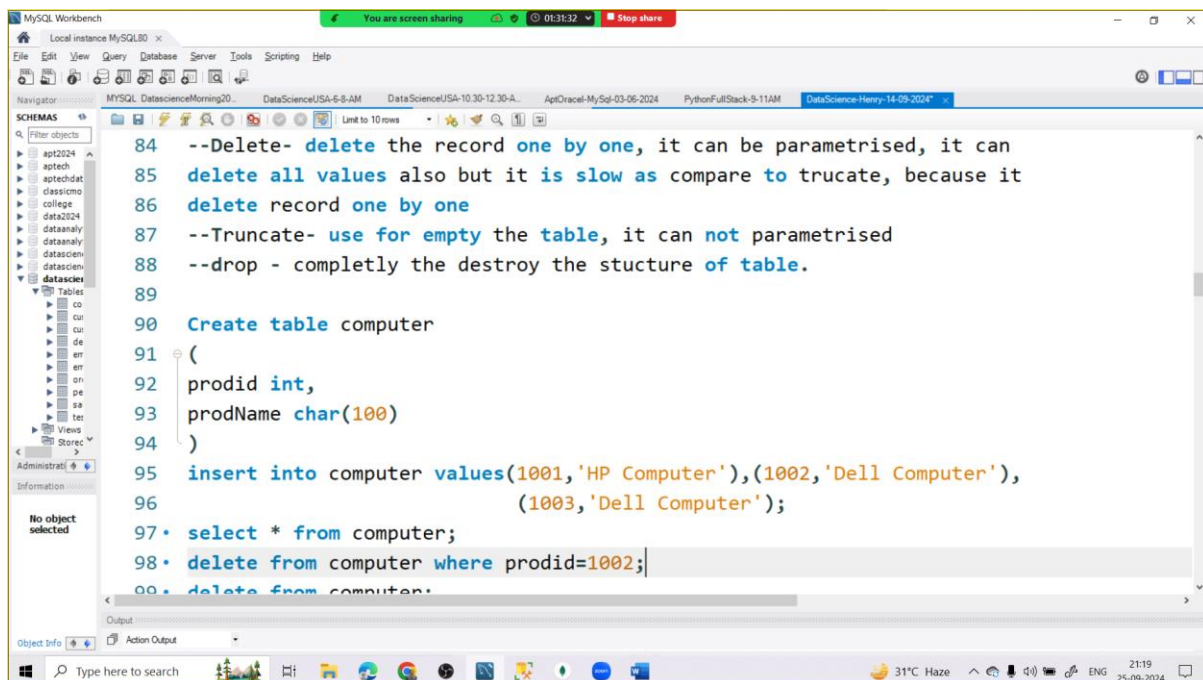
```
58 • create table empdata
59 (
60     empid int primary key not null,
61     empname char(100) not null,
62     emplicence char(50) unique key,
63     emppassport char(70) unique key
64 )
65 select * from empdata;
66 • insert into empdata values(001,'James','LC001','PS001');
67
68 • create table Department
69 (
70     empid int,
71     foreign key (empid) references empdata(empid),
72     deptname varchar(100)
73 )
```

The interface includes a Navigator pane on the left showing a schema with various tables like 'ap12024', 'aptech', 'classico', etc. The bottom status bar shows the system time as 21:18 on 25-09-2024.



This screenshot shows the MySQL Workbench interface with a SQL editor. The left sidebar displays a schema tree with various databases like 'ap12024', 'aptech', and 'data2024'. The main editor contains SQL code for creating a foreign key, inserting data into 'department' and 'empdata' tables, and selecting records. Comments explain the differences between deleting one record at a time versus truncating the entire table.

```
71 foreign key (empid) references empdata(empid),
72 deptname varchar(100)
73 );
74 insert into department values(001, 'IT Dept');
75 select * from department;
76
77
78 insert into empdata values(002, 'Kellon', 'LC002', 'PS002'),
79                        (003, 'Bob', 'LC003', 'PS003');
80
81 select * from empdata where empid=2;
82
83 --Delete-Drop-Truncate-
84 --Delete- delete the record one by one, it can be parametrised, it can
85 delete all values also but it is slow as compare to truncate, because it
86 delete record one by one
```



This screenshot shows the MySQL Workbench interface with a SQL editor. The left sidebar displays a schema tree. The main editor contains SQL code for creating a 'computer' table, inserting data, and deleting a specific record. Comments explain the differences between deleting one record at a time versus truncating the entire table.

```
84 --Delete- delete the record one by one, it can be parametrised, it can
85 delete all values also but it is slow as compare to truncate, because it
86 delete record one by one
87 --Truncate- use for empty the table, it can not parametrised
88 --drop - completely the destroy the stucture of table.
89
90 Create table computer
91 (
92   prodid int,
93   prodName char(100)
94 )
95 insert into computer values(1001, 'HP Computer'), (1002, 'Dell Computer'),
96                        (1003, 'Dell Computer');
97 select * from computer;
98 delete from computer where prodid=1002;
99 delete from computer;
```

This screenshot shows the MySQL Workbench interface with a SQL editor containing the following queries:

```
97 • select * from computer;
98 • delete from computer where prodid=1002;
99 • delete from computer;
100
101 • truncate computer;
102
103 • drop table computer;
104
105 • --update the values from table
106 update computer set prodName='Apple Computer' where prodid=1003;
107 • select prodName from computer;
108 • select * from computer;
109 • --distinct show unique record
110 select distinct * from computer;
111
112 • --order by table
```

The interface includes a left-hand sidebar with a 'SCHEMAS' tree, a top menu bar, and a bottom status bar showing system information like temperature and time.

This screenshot shows the MySQL Workbench interface with a SQL editor containing the following queries:

```
110 select distinct * from computer;
111
112 • --order by table
113 select * from computer order by prodid desc;
114
115 • --display top record using by limit
116 select * from computer limit 2;
117
118 • --Wild Card using by like
119 select * from computer where prodName like 'D%';
120
121 • --In Operator
122 select * from computer where prodName in ('HP Computer','Apple Computer');
123
124 • select * from computer where prodid between 1001 and 1002;
125
```

The interface is consistent with the first screenshot, showing the same MySQL Workbench environment with various toolbars and a sidebar.

This screenshot shows the MySQL Workbench interface with a SQL editor containing the following queries:

```
124 • select * from computer where prodid between 1001 and 1002;
125
126 • --import data
127 select * from salaries;
128 • select * from salaries where salary between 60000 and 100000;
129
130 • --Joins in SQL--
131 select * from person;
132 • select * from orders;
133
134 • --Inner join
135 select person.p_id,person.firstName,person.city,orders.orderNo from person
136 inner join orders on person.p_id=orders.p_id;
137
138 • select * from person
139 inner join orders on person.p_id=orders.p_id;
```

The interface includes a Navigator on the left showing a database schema with tables like 'computer', 'salaries', 'person', and 'orders'. The bottom status bar shows the system time as 21:19 on 25-09-2024.

This screenshot shows the MySQL Workbench interface with a SQL editor containing the following queries:

```
138 • select * from person
139 inner join orders on person.p_id=orders.p_id;
140
141 • --Left Join
142 select * from person
143 left join orders on person.p_id=orders.p_id;
144
145 • --Right Join
146 select * from person
147 Right join orders on person.p_id=orders.p_id;
148
149 • --Full outer join
150 select * from person left join orders on person.p_id=orders.p_id
151 union all
152 select * from person Right join orders on person.p_id=orders.p_id;
153
```

The interface includes a Navigator on the left showing a database schema with tables like 'person' and 'orders'. The bottom status bar shows the system time as 21:20 on 25-09-2024.

MySQL Workbench

You are screen sharing 01:32:43 Stop share

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator MySQL DatascienceMorning20... DataScienceUSA-6-8-AM DataScienceUSA-10-30-12-30-A... AptOracle-MySQL-03-06-2024 PythonFullStack-9-11AM DataScience-Henry-14-09-2024

SCHEMAS Filter objects

apt2024 apttech apttechdat classicmo college data2024 dataanaly dataanali datasocien datasocien

Tables co cur cur de em on pe sa ter

Views Storec

Administrat

Information

No object selected

```
152 select * from person Right join orders on person.p_id=orders.p_id;
153
154 --cross join
155 select * from person cross join orders;
156
157 --union show the unique record
158 select * from customers1
159 union
160 select * from customers2;
161
162 --union all show all values or also show duplicate values
163 select * from customers1
164 union all
165 select * from customers2;
166
167
```

Object Info

Type here to search

31°C Haze 21:20 25-09-2024

MySQL Workbench

You are screen sharing 01:32:54 Stop share

Local instance MySQL80

File Edit View Query Database Server Tools Scripting Help

Navigator MySQL DatascienceMorning20... DataScienceUSA-6-8-AM DataScienceUSA-10-30-12-30-A... AptOracle-MySQL-03-06-2024 PythonFullStack-9-11AM DataScience-Henry-14-09-2024

SCHEMAS Filter objects

apt2024 apttech apttechdat classicmo college data2024 dataanaly dataanali datasocien datasocien

Tables co cur cur de em on pe sa ter

Views Storec

Administrat

Information

No object selected

```
167
168 --View-it is the temporary table
169
170 select * from Salaries;
171
172 create view vSalary as
173 select * from salaries where Salary>=100000;
174
175 select * from vSalary;
176
177 --Date Function--
178 select now();
179 select curdate();
180 select curtime();
181 select date(now());
182
```

Object Info

Type here to search

31°C Haze 21:20 25-09-2024

This screenshot shows the MySQL Workbench interface with a SQL editor containing several queries. The left sidebar shows a schema named 'datascience' with various tables. The main editor area displays the following SQL code:

```
182
183 --group by the data according to particular column
184 select * from salaries;
185 • select position,sum(salary) from salaries group by position;
186
187 • --after group by show record only those salary greater 100000
188
189 select position,sum(salary) from salaries group by position having
190 sum(salary)>100000;
191
192 • --order by use for arrange the data
193
194 select * from salaries order by salary desc;
195 • select * from salaries order by salary asc;
196
197
```

The bottom status bar indicates the system temperature is 31°C, it is hazy, and the date is 25-09-2024.

This screenshot shows the MySQL Workbench interface with a SQL editor containing queries related to transaction properties and table creation. The left sidebar shows the same schema as the previous screenshot. The main editor area displays the following SQL code:

```
197
198 --Trasaction- to run all process into single unit if any one process is
199 --failure so all process rollback
200 --ACID Property
201 1)-Automicity- all process run automatic
202 2)-Consistency- All Process run into the consistant way
203 3) Isolation- all process run independently
204 4) Duration- to complete the process within duration
205
206 use datascience2024;
207 • Create table mytranData
208 (
209 prodid int,
210 prodName char(100)
211 );
212 • insert into mytranData values(1001,'HP Computer'),(1002,'Dell Computer'),
```

The bottom status bar indicates the system temperature is 31°C, it is hazy, and the date is 25-09-2024.

This screenshot shows the MySQL Workbench interface with a SQL editor containing the following queries:

```
209 prodid int,  
210 prodName char(100)  
211 );  
212 • insert into mytranData values(1001,'HP Computer'),(1002,'Dell Computer'),  
213      (1003,'Dell Computer');  
214 • select * from mytranData;  
215  
216 • start transaction;  
217 • delete from mytranData where prodid=1001;  
218  
219 • --rollback use for undo  
220 rollback;  
221  
222 • --When we want to save permanently use commit  
223  
224 start transaction;
```

The interface includes a Navigator pane on the left showing a schema named 'dataScienceMorning20' with various tables. The bottom status bar indicates the system time is 21:20 on 25-09-2024.

This screenshot shows the MySQL Workbench interface with a SQL editor containing the following queries:

```
221  
222 • --When we want to save permanently use commit  
223  
224 start transaction;  
225 • delete from mytranData where prodid=1001;  
226 • commit;  
227  
228 • --Once you commit after that you can not rollback  
229 rollback;  
230  
231 • select * from mytranData;  
232  
233  
234  
235  
236
```

The interface is consistent with the previous screenshot, showing the same schema and table structure. The bottom status bar indicates the system time is 21:21 on 25-09-2024.