

# *AIRPORT*

# *DATABASE*

# **INTRODUCTION**

*An airport database schema is designed to manage and organize the vast amounts of information associated with airport operations, including flights, airlines, passengers, and ticketing. It is a structured system that enables efficient data storage, retrieval, and management, supporting various functionalities like scheduling, passenger tracking, and airline coordination.*

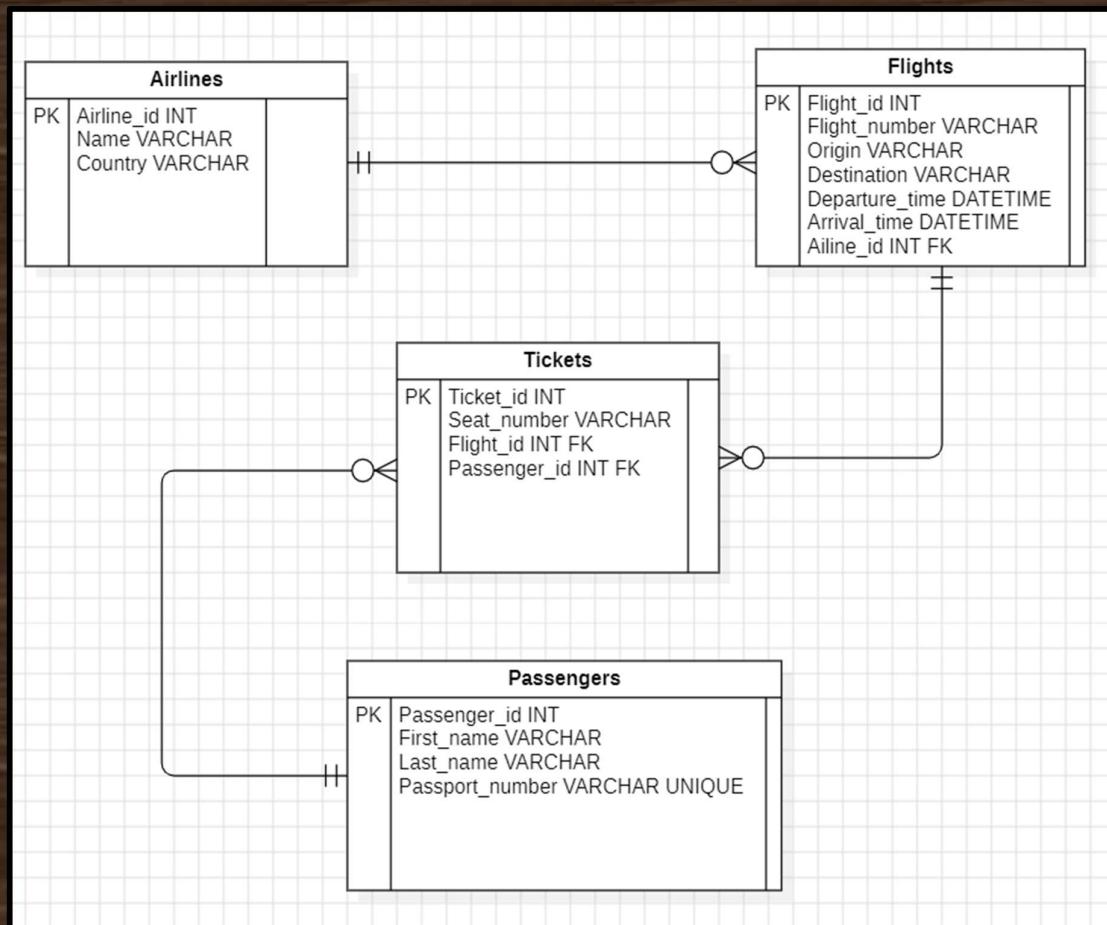
*In an airport setting, data is highly interconnected. Airlines operate multiple flights, each flight carries numerous passengers, and each passenger holds a unique ticket. The schema aims to capture these relationships by defining tables for key entities (such as airlines, flights, passengers, and tickets) and establishing links through primary and foreign keys.*

*This relational model enables easy access to critical information. For example, it allows airport staff to:*

- Retrieve schedules for flights operated by a specific airline.*
- Track all passengers booked on a particular flight.*
- Manage ticket information and ensure secure passenger records.*

*The primary objective of this schema is to create an efficient, scalable structure that supports complex airport operations, enhances data accuracy, and ensures seamless access to information across various departments and systems within the airport.*

# *ER DIAGRAM*



# ***DATABASE DESIGN***

*Database: Airport*

*Tables:*

- *Airlines*
- *Flights*
- *Passengers*
- *Tickets*

# *DATA DEFINITION LANGUAGE (DDL)*

## **CREATING DATABASE AND TABLES:**

### *a) Creating Database*

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the command `create database Airport;` is entered. The output pane shows the result of the command:

#	Time	Action	Message
1	19:22:43	create database Airport	1 row(s) affected

### *b) Use Database*

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the command `use Airport;` is entered. The output pane shows the result of the command:

#	Time	Action	Message
1	19:22:43	create database Airport	1 row(s) affected
2	19:25:27	use Airport	0 row(s) affected

### *c) Creating table (Airlines)*

The screenshot shows the MySQL Workbench interface. In the SQL editor tab, the command to create a table is entered:

```
1 • CREATE TABLE Airlines (
 2     airline_id INT PRIMARY KEY,
 3     name VARCHAR(100),
 4     country VARCHAR(50)
 5 );
 6
```

The output pane shows the result of the command:

#	Time	Action	Message
1	19:46:11	CREATE TABLE Airlines ( airline_id INT PRIMARY KEY, name VARCHAR(100), country VARCHAR(50) )	0 row(s) affected

d) Creating table (Flights)

The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code in the editor is:

```
1 • CREATE TABLE Flights (
2     flight_id INT PRIMARY KEY,
3     airline_id INT,
4     flight_number VARCHAR(10),
5     origin VARCHAR(50),
6     destination VARCHAR(50),
7     departure_time DATETIME,
8     arrival_time DATETIME,
9     FOREIGN KEY (airline_id) REFERENCES Airlines(airline_id)
10 );
```

The output pane shows a single log entry:

#	Time	Action	Message
1	19:49:11	CREATE TABLE Flights ( flight_id INT PRIMARY KEY, airline_id INT, flight_number VARCHAR(10), ori... )	0 row(s) affected

e) Creating table (Passengers)

The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code in the editor is:

```
1 • CREATE TABLE Passengers (
2     passenger_id INT PRIMARY KEY,
3     first_name VARCHAR(50),
4     last_name VARCHAR(50),
5     passport_number VARCHAR(15) UNIQUE
6 );
7
```

The output pane shows a single log entry:

#	Time	Action	Message
1	19:51:18	CREATE TABLE Passengers ( passenger_id INT PRIMARY KEY, first_name VARCHAR(50), last_name V... )	0 row(s) affected

f) Creating table (Tickets)

The screenshot shows the MySQL Workbench interface with the SQL editor tab active. The code in the editor is:

```
1 • CREATE TABLE Tickets (
2     ticket_id INT PRIMARY KEY,
3     flight_id INT,
4     passenger_id INT,
5     seat_number VARCHAR(5),
6     FOREIGN KEY (flight_id) REFERENCES Flights(flight_id),
7     FOREIGN KEY (passenger_id) REFERENCES Passengers(passenger_id)
8 );
```

The output pane shows a single log entry:

#	Time	Action	Message
1	19:52:23	CREATE TABLE Tickets ( ticket_id INT PRIMARY KEY, flight_id INT, passenger_id INT, seat_number ... )	0 row(s) affected

## To Structure Of Table:

### 1. Airlines

43 • desc Airlines;

Field	Type	Null	Key	Default	Extra
airline_id	int	NO	PRI	NULL	
name	varchar(100)	YES		NULL	
country	varchar(50)	YES		NULL	

### 2. Flights

47 • desc Flights;

48

Field	Type	Null	Key	Default	Extra
flight_id	int	NO	PRI	NULL	
airline_id	int	YES	MUL	NULL	
flight_number	varchar(10)	YES		NULL	
origin	varchar(50)	YES		NULL	
destination	varchar(50)	YES		NULL	

### 3. Passengers

```
44 • desc Passengers;
```

45

---

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	passenger_id	int	NO	PRI	NULL	
	first_name	varchar(50)	YES		NULL	
	last_name	varchar(50)	YES		NULL	
	passport_number	varchar(15)	YES	UNI	NULL	

### 4. Tickets

```
46 • desc Tickets;
```

47

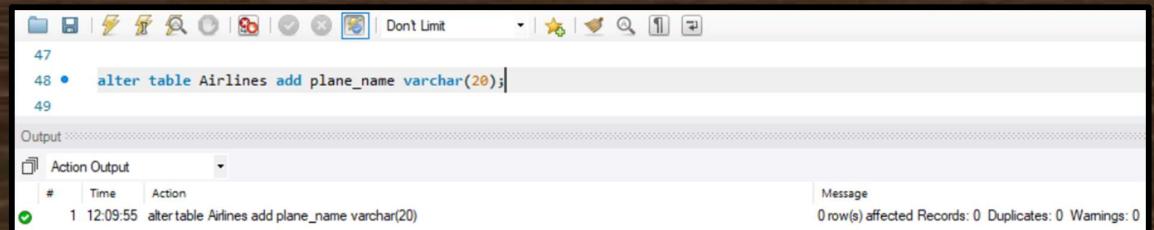
---

Result Grid | Filter Rows: Export: Wrap Cell Content:

	Field	Type	Null	Key	Default	Extra
▶	ticket_id	int	NO	PRI	NULL	
	flight_id	int	YES	MUL	NULL	
	passenger_id	int	YES	MUL	NULL	
	seat_number	varchar(5)	YES		NULL	

## I. ALTER

### 1. Alter Table Add Column



The screenshot shows the MySQL Workbench interface. In the SQL tab, the command `alter table Airlines add plane_name varchar(20);` is entered. In the Output tab, the results show a single row: # 1, Time 12:09:55, Action alter table Airlines add plane\_name varchar(20). The Message panel indicates 0 row(s) affected, 0 Records: 0, 0 Duplicates: 0, and 0 Warnings: 0.

```
47
48 • alter table Airlines add plane_name varchar(20);
49

Output:
Action Output
# Time Action
1 12:09:55 alter table Airlines add plane_name varchar(20)

Message
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

### 2. Alter Table Modify Column



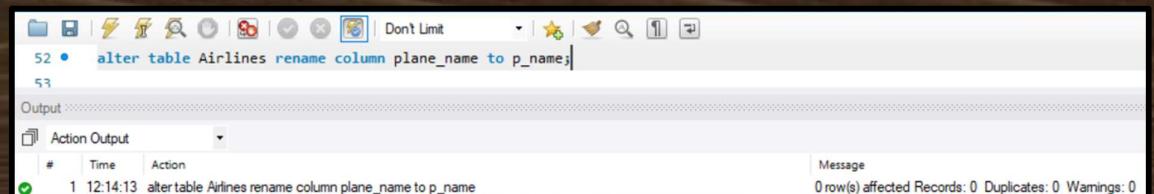
The screenshot shows the MySQL Workbench interface. In the SQL tab, the command `alter table Airlines modify plane_name varchar(100);` is entered. In the Output tab, the results show a single row: # 1, Time 12:11:21, Action altertable Airlines modify plane\_name varchar(100). The Message panel indicates 0 row(s) affected, 0 Records: 0, 0 Duplicates: 0, and 0 Warnings: 0.

```
49
50 • alter table Airlines modify plane_name varchar(100);
51

Output:
Action Output
# Time Action
1 12:11:21 altertable Airlines modify plane_name varchar(100)

Message
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

### 3. Alter Table Rename Column



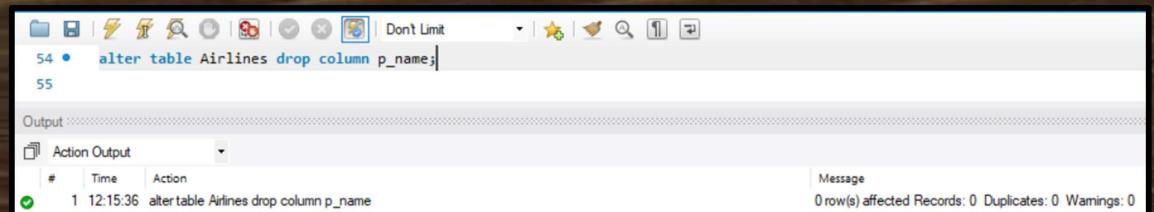
The screenshot shows the MySQL Workbench interface. In the SQL tab, the command `alter table Airlines rename column plane_name to p_name;` is entered. In the Output tab, the results show a single row: # 1, Time 12:14:13, Action altertable Airlines rename column plane\_name to p\_name. The Message panel indicates 0 row(s) affected, 0 Records: 0, 0 Duplicates: 0, and 0 Warnings: 0.

```
52 • alter table Airlines rename column plane_name to p_name;
53

Output:
Action Output
# Time Action
1 12:14:13 altertable Airlines rename column plane_name to p_name

Message
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

### 4. Alter Table Drop Column



The screenshot shows the MySQL Workbench interface. In the SQL tab, the command `alter table Airlines drop column p_name;` is entered. In the Output tab, the results show a single row: # 1, Time 12:15:36, Action altertable Airlines drop column p\_name. The Message panel indicates 0 row(s) affected, 0 Records: 0, 0 Duplicates: 0, and 0 Warnings: 0.

```
54 • alter table Airlines drop column p_name;
55

Output:
Action Output
# Time Action
1 12:15:36 altertable Airlines drop column p_name

Message
0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0
```

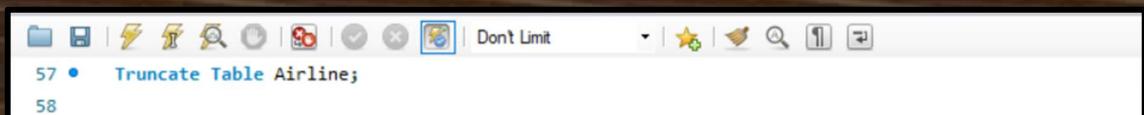
## *II. Drop*



A screenshot of a MySQL Workbench interface. The toolbar at the top includes icons for file operations, search, and connection management. The main window shows a query editor with the following content:

```
58 •  drop Table Airlines;
59
```

## *III. Truncate*



A screenshot of a MySQL Workbench interface. The toolbar at the top includes icons for file operations, search, and connection management. The main window shows a query editor with the following content:

```
57 •  Truncate Table Airline;
58
```

## DATA MANIPULATION LANGUAGE (DML)

### I. {INSERT INTO} Passengers:

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is an `INSERT INTO` statement for the `Flights` table, containing 20 rows of flight data. Below the code, the output pane shows the execution message: "1 12:45:00 INSERT INTO Flights ... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0".

```
79 •    INSERT INTO Flights (flight_id, airline_id, flight_number, origin, destination, departure_time, arrival_time) VALUES
80      (101, 1, 'AA101', 'New York', 'Los Angeles', '2023-11-01 08:00:00', '2023-11-01 11:00:00'),
81      (102, 2, 'DL202', 'Los Angeles', 'Chicago', '2023-11-01 09:00:00', '2023-11-01 12:00:00'),
82      (103, 3, 'UA303', 'Chicago', 'Houston', '2023-11-01 10:00:00', '2023-11-01 13:00:00'),
83      (104, 4, 'LH404', 'Frankfurt', 'New York', '2023-11-01 11:00:00', '2023-11-01 14:00:00'),
84      (105, 5, 'BA505', 'London', 'Paris', '2023-11-01 12:00:00', '2023-11-01 13:30:00'),
85      (106, 6, 'AF606', 'Paris', 'Rome', '2023-11-01 13:00:00', '2023-11-01 14:30:00'),
86      (107, 7, 'EK707', 'Dubai', 'Sydney', '2023-11-01 14:00:00', '2023-11-01 22:00:00'),
87      (108, 8, 'QR808', 'Doha', 'Bangkok', '2023-11-01 15:00:00', '2023-11-01 19:00:00'),
88      (109, 9, 'SQ909', 'Singapore', 'Tokyo', '2023-11-01 16:00:00', '2023-11-01 20:00:00'),
89      (110, 10, 'AC1010', 'Toronto', 'New York', '2023-11-01 17:00:00', '2023-11-01 19:00:00'),
90      (111, 11, 'QF111', 'Sydney', 'Los Angeles', '2023-11-01 18:00:00', '2023-11-01 08:00:00'),
91      (112, 12, 'TK1212', 'Istanbul', 'Berlin', '2023-11-01 19:00:00', '2023-11-01 21:00:00'),
92      (113, 13, 'KL1313', 'Amsterdam', 'Dubai', '2023-11-01 20:00:00', '2023-11-01 23:00:00'),
93      (114, 14, 'LX1414', 'Zurich', 'London', '2023-11-01 21:00:00', '2023-11-01 22:00:00'),
94      (115, 15, 'JL1515', 'Tokyo', 'Beijing', '2023-11-01 22:00:00', '2023-11-01 23:30:00'),
95      (116, 16, 'CA1616', 'Beijing', 'Seoul', '2023-11-01 23:00:00', '2023-11-02 01:00:00'),
96      (117, 17, 'KE1717', 'Seoul', 'Shanghai', '2023-11-02 00:00:00', '2023-11-02 02:00:00'),
97      (118, 18, 'SU1818', 'Moscow', 'Paris', '2023-11-02 01:00:00', '2023-11-02 04:00:00'),
98      (119, 19, 'IB1919', 'Madrid', 'Lisbon', '2023-11-02 02:00:00', '2023-11-02 03:00:00'),
99      (120, 20, 'AZ2020', 'Rome', 'Athens', '2023-11-02 03:00:00', '2023-11-02 04:30:00);
```

Output:

#	Time	Action
1	12:45:00	INSERT INTO Flights flight_id, airline_id, flight_number, origin, destination, departure_time, arrival_time) VALUES ... 20 row(s) affected Records: 20 Duplicates: 0 Warnings: 0

### II. {UPDATE} With Set & Where Clause:

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is an `update` statement for the `Passengers` table, setting `first_name` to "Alice" where `passenger_id` is 1001. Below the code, the output pane shows the execution message: "1 17:54:08 update Passengers set first\_name='Alice' where passenger\_id=1001 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0".

```
146 •    update Passengers set first_name="Alice" where passenger_id=1001;
```

Output:

#	Time	Action
1	17:54:08	update Passengers set first_name='Alice' where passenger_id=1001 1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0

### III. {DELETE FROM} With Where Clause:

The screenshot shows the MySQL Workbench interface with a query editor window. The code entered is a `delete from` statement for the `Tickets` table, where `ticket_id` is 19. Below the code, the output pane shows the execution message: "1 18:00:58 delete from Tickets where ticket\_id=19 1 row(s) affected".

```
149 •    delete from Tickets where ticket_id=19;
```

Output:

#	Time	Action
1	18:00:58	delete from Tickets where ticket_id=19 1 row(s) affected

## DATA QUERY LANGUAGE (DQL)

### I. {SELECT Command}:

- To Display all records in a table:

The screenshot shows the MySQL Workbench interface. At the top, there is a toolbar with various icons. Below the toolbar, the SQL editor window displays the following code:

```
151 •  select * from flights;
```

The Result Grid below shows the output of the query:

flight_id	airline_id	flight_number	origin	destination	departure_time	arrival_time
101	1	AA101	New York	Los Angeles	2023-11-01 08:00:00	2023-11-01 11:00:00
102	2	DL202	Los Angeles	Chicago	2023-11-01 09:00:00	2023-11-01 12:00:00
103	3	UA303	Chicago	Houston	2023-11-01 10:00:00	2023-11-01 13:00:00
104	4	LH404	Frankfurt	New York	2023-11-01 11:00:00	2023-11-01 14:00:00
105	5	BA505	London	Paris	2023-11-01 12:00:00	2023-11-01 13:30:00
106	6	AF606	Paris	Rome	2023-11-01 13:00:00	2023-11-01 14:30:00
107	7	EK707	Dubai	Sydney	2023-11-01 14:00:00	2023-11-01 22:00:00
108	8	QR808	Doha	Bangkok	2023-11-01 15:00:00	2023-11-01 19:00:00
109	9	SQ909	Singapore	Tokyo	2023-11-01 16:00:00	2023-11-01 20:00:00
110	10	AC1010	Toronto	New York	2023-11-01 17:00:00	2023-11-01 19:00:00
111	11	QF111	Sydney	Los Angeles	2023-11-01 18:00:00	2023-11-01 08:00:00
112	12	TK1212	Istanbul	Berlin	2023-11-01 19:00:00	2023-11-01 21:00:00
113	13	KL1313	Amsterdam	Dubai	2023-11-01 20:00:00	2023-11-01 23:00:00
114	14	LX1414	Zurich	London	2023-11-01 21:00:00	2023-11-01 22:00:00
115	15	JL1515	Tokyo	Beijing	2023-11-01 22:00:00	2023-11-01 23:30:00
116	16	CA1616	Beijing	Seoul	2023-11-01 23:00:00	2023-11-02 01:00:00
117	17	KE1717	Seoul	Shanghai	2023-11-02 00:00:00	2023-11-02 02:00:00
118	18	SU1818	Moscow	Paris	2023-11-02 01:00:00	2023-11-02 04:00:00
119	19	IB1919	Madrid	Lisbon	2023-11-02 02:00:00	2023-11-02 03:00:00
120	20	AZ2020	Rome	Athens	2023-11-02 03:00:00	2023-11-02 04:30:00
HULL	HULL	HULL	HULL	HULL	HULL	HULL

The Output pane at the bottom shows the following information:

#	Time	Action
1	18:09:19	select * from flights

Message: 20 row(s) returned

- *SELECT command with WHERE Clause:*

The screenshot shows a MySQL Workbench interface. At the top, there's a toolbar with various icons. Below the toolbar, the SQL editor pane displays the following code:

```
153 •  select * from flights where destination="Dubai";
154
155
156
157
```

Below the code, the "Result Grid" shows a single row of data:

	flight_id	airline_id	flight_number	origin	destination	departure_time	arrival_time
▶	113	13	KL1313	Amsterdam	Dubai	2023-11-01 20:00:00	2023-11-01 23:00:00
*	NULL	NULL	NULL	NULL	NULL	NULL	NULL

At the bottom, the "Output" pane shows the log entry:

Action Output
# Time Action
1 18:11:47 select * from flights where destination="Dubai"

On the right side of the output pane, there is a "Message" section with the text "1 row(s) returned".

## ***II. {SHOW Command}:***

- *To display all tables in a database:*

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the query `show tables;` and its execution history from line 155 to 156.
- Result Grid:** Shows the output of the query as a tree view of tables in the `airport` database:
  - Tables\_in\_airport
  - airlines
  - flights
  - passengers
  - tickets
- Action Output:** A log of actions taken by the user, showing the execution of the `show tables` command at 18:16:11.
- Message:** Indicated that 4 row(s) were returned.

- *To display databases:*

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the query `show databases;` and its execution history from line 156 to 160.
- Result Grid:** Shows the output of the query as a tree view of databases:
  - Database
  - airport
  - batch3to5
  - college
  - information\_schema
  - mysql
  - performance\_schema
  - pizza\_sales\_analysis
  - sql1
  - sys
- Action Output:** A log of actions taken by the user, showing the execution of the `show databases` command at 18:16:50.
- Message:** Indicated that 9 row(s) were returned.

# *LIKE Query*

- *Using “%”*

A screenshot of the MySQL Workbench interface. The SQL editor window shows the following query:

```
158 •  select * from Passengers where First_name like 'J%';  
159  
160  
161  
162
```

The Result Grid displays the following data:

	passenger_id	first_name	last_name	passport_number
▶	1002	Jane	Smith	P23456789
▶	1011	Jack	Moore	P12345098
*	HULL	HULL	HULL	HULL

The Output pane shows the query and its execution details:

#	Time	Action
1	18:23:43	select * from Passengers where First_name like 'J%'

Message: 2 row(s) returned

- *Using “\_”*

A screenshot of the MySQL Workbench interface. The SQL editor window shows the following query:

```
160 •  select * from Passengers where Last_name like 'D_e';  
161  
162
```

The Result Grid displays the following data:

	passenger_id	first_name	last_name	passport_number
▶	1001	Alice	Doe	P12345678
*	HULL	HULL	HULL	HULL

The Output pane shows the query and its execution details:

#	Time	Action
1	18:25:30	select * from Passengers where Last_name like 'D_e'

Message: 1 row(s) returned

# *LIMIT Clause*

The screenshot shows a MySQL Workbench interface. The SQL editor window contains the following code:

```
162 •  select * from Tickets limit 5;
```

The results grid displays the following data:

	ticket_id	flight_id	passenger_id	seat_number
▶	1	101	1001	1A
	2	101	1002	1B
	3	102	1003	2A
	4	102	1004	2B
	5	103	1005	3A

The status bar at the bottom right indicates "5 row(s) returned".

# *GROUP BY Clause*

## a) GROUP BY:

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
164 •  select passenger_id, concat(First_name, ' ', Last_name) as Full_name
165   from passengers
166   group by passenger_id;
167
168
```

The results grid displays the following data:

passenger_id	Full_name
1001	Alice Doe
1002	Jane Smith
1003	Alice Johnson
1004	Bob Brown
1005	Chris Davis
1006	Emily Wilson
1007	Frank Miller
1008	Grace Taylor
1009	Henry Anderson

The output pane shows the query and its execution details:

```
1 18:36:14 select passenger_id, concat(First_name,' ',Last_name) as Full_name from passengers group by passenger_id
Message
20 row(s) returned
```

## b) GROUP BY with HAVING Clause:

The screenshot shows the MySQL Workbench interface with a query editor and results grid. The query is:

```
168 •  select flight_id, flight_number, origin, destination
169   from Flights
170   group by flight_id
171   having destination="London";
172
```

The results grid displays the following data:

flight_id	flight_number	origin	destination
114	LX1414	Zurich	London
HULL	HULL	HULL	HULL

The output pane shows the query and its execution details:

```
1 18:39:43 select flight_id, flight_number, origin, destination from Flights group by flight_id having destination="London"
Message
1 row(s) returned
```

# *ORDER BY Clause*

## a) ORDER BY ASC:

The screenshot shows the MySQL Workbench interface with a query editor and a results grid.

Query Editor (Text):

```
173 •  select * from Flights
174      order by destination;
```

Result Grid (Table):

	flight_id	airline_id	flight_number	origin	destination	departure_time	arrival_time
▶	120	20	AZ2020	Rome	Athens	2023-11-02 03:00:00	2023-11-02 04:30:00
	108	8	QR808	Doha	Bangkok	2023-11-01 15:00:00	2023-11-01 19:00:00
	115	15	JL1515	Tokyo	Beijing	2023-11-01 22:00:00	2023-11-01 23:30:00
	112	12	TK1212	Istanbul	Berlin	2023-11-01 19:00:00	2023-11-01 21:00:00
	102	2	DL202	Los Angeles	Chicago	2023-11-01 09:00:00	2023-11-01 12:00:00
	113	13	KL1313	Amsterdam	Dubai	2023-11-01 20:00:00	2023-11-01 23:00:00
	103	3	UA303	Chicago	Houston	2023-11-01 10:00:00	2023-11-01 13:00:00
	119	19	IB1919	Madrid	Lisbon	2023-11-02 02:00:00	2023-11-02 03:00:00
	114	14	LX1414	Zurich	London	2023-11-01 21:00:00	2023-11-01 22:00:00
	101	1	AA101	New York	Los Angeles	2023-11-01 08:00:00	2023-11-01 11:00:00
	111	11	QF111	Sydney	Los Angeles	2023-11-01 18:00:00	2023-11-01 08:00:00
	104	4	LH404	Frankfurt	New York	2023-11-01 11:00:00	2023-11-01 14:00:00
	110	10	AC1010	Toronto	New York	2023-11-01 17:00:00	2023-11-01 19:00:00
	105	5	BA505	London	Paris	2023-11-01 12:00:00	2023-11-01 13:30:00
	118	18	SU1818	Moscow	Paris	2023-11-02 01:00:00	2023-11-02 04:00:00
	106	6	AF606	Paris	Rome	2023-11-01 13:00:00	2023-11-01 14:30:00
	116	16	CA1616	Beijing	Seoul	2023-11-01 23:00:00	2023-11-02 01:00:00
	117	17	KE1717	Seoul	Shanghai	2023-11-02 00:00:00	2023-11-02 02:00:00
	107	7	EK707	Dubai	Sydney	2023-11-01 14:00:00	2023-11-01 22:00:00
	109	9	SQ909	Singapore	Tokyo	2023-11-01 16:00:00	2023-11-01 20:00:00
*	HULL	HULL	HULL	HULL	HULL	HULL	HULL

Output:

Action Output

#	Time	Action
1	19:09:00	select * from Flights order by destination

Message: 20 row(s) returned

**b) ORDER BY DESC:**

```
176 •  select * from Passengers  
177      order by first_name desc;  
178  
179  
180
```

Result Grid | Filter Rows:  Edit: Export/Import: Wrap Cell Content:

	passenger_id	first_name	last_name	passport_number
▶	1020	Sam	Allen	P01232109
	1019	Ruby	Hall	P90103210
	1018	Quinn	Walker	P89004321
	1017	Paul	Lewis	P78905432
	1016	Oscar	Clark	P67806543
	1015	Nina	Harris	P56707654
	1014	Mia	White	P45608765
	1013	Leo	Jackson	P34509876
	1012	Karen	Martin	P23450987
	1002	Jane	Smith	P23456789
	1011	Jack	Moore	P12345098
	1010	Ivy	Thomas	P01234567
	1009	Henry	Anderson	P90123456
	1008	Grace	Taylor	P89012345
	1007	Frank	Miller	P78901234
	1006	Emily	Wilson	P67890123
	1005	Chris	Davis	P56789012
	1004	Bob	Brown	P45678901
	1001	Alice	Doe	P12345678
	1003	Alice	Johnson	P34567890
*	NULL	NULL	NULL	NULL

Passengers 13 ×

Output:

Action Output

#	Time	Action
1	19:11:19	select * from Passengers order by first_name desc

Message  
20 row(s) returned

# BUILT IN FUNCTIONS

## a) CONCAT:

```
179 •   select concat(First_name, ' ',Last_name) as Full_name from passengers;
180
181
182
183
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

Full_name
Alice Doe
Jane Smith
Alice Johnson
Bob Brown
Chris Davis
Emily Wilson

Result 14 x

Output

Action Output

#	Time	Action
1	20:58:23	select concat(First_name,' ',Last_name) as Full_name from passengers

Message  
20 row(s) returned

## b) LOWER:

```
181 •   select flight_id,lower(destination) as destination from flights;
182
183
184
185
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

flight_id	destination
101	los angeles
102	chicago
103	houston
104	new york
105	paris
106	rome

Result 15 x

Output

Action Output

#	Time	Action
1	21:00:50	select flight_id.lower(destination) as destination from flights

Message  
20 row(s) returned

c) **UPPER:**

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the query: `select flight_number,upper(origin) as origin from flights;`
- Result Grid:** Displays the results of the query in a tabular format:

flight_number	origin
AA101	NEW YORK
DL202	LOS ANGELES
UA303	CHICAGO
LH404	FRANKFURT
BA505	LONDON
AF606	PARIS

- Action Output:** Shows the executed query: `select flight_number.upper(origin) as origin from flights`.
- Message:** `20 row(s) returned`

d) **REPLACE:**

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Contains the query: `select replace('WELCOME HOME','HOME','HOUSE') as REPLACED_STRING;`
- Result Grid:** Displays the results of the query in a tabular format:

REPLACED_STRING
WELCOME HOUSE

- Action Output:** Shows the executed query: `select replace('WELCOME HOME','HOME','HOUSE') as REPLACED_STRING`.
- Message:** `1 row(s) returned`

e) **REVERSE:**

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the query: `select reverse(first_name) from passengers;`
- Result Grid:** Shows the results of the query, listing names reversed:

reverse(first_name)
ecilA
enaJ
ecilA
boB
sirhC
ylimE
knarF
ecarG
- Action Output:** Shows the executed query and the message "20 row(s) returned".

#	Time	Action
1	21:08:28	select reverse(first_name) from passengers

f) **LENGTH:**

The screenshot shows the MySQL Workbench interface with the following details:

- SQL Editor:** Displays the query: `select length(origin) from flights;`
- Result Grid:** Shows the results of the query, listing the lengths of origin cities:

length(origin)
8
11
7
9
6
5
5
4
- Action Output:** Shows the executed query and the message "20 row(s) returned".

#	Time	Action
1	21:09:53	select length(origin) from flights

### g) SUBSTRING:

The screenshot shows the MySQL Workbench interface with the following details:

- Query Editor:** Displays the SQL command: `select substr(last_name,1,3) as extracted_string from passengers;`
- Result Grid:** Shows the output of the query in a grid format. The column is labeled "extracted\_string" and contains the following values:
  - Doe
  - Smi
  - Joh
  - Bro
  - Dav
  - Wil
  - Mil
  - Tay
- Action Output:** A log of the action taken.

#	Time	Action	Message
1	21:11:57	select substr(last_name,1,3) as extracted_string from passengers	20 row(s) returned

# UNION OPERATIONS

The screenshot shows a MySQL Workbench interface with a query editor and results pane.

Query:

```
192
193 • select airline_id as id, name as identity, country as Nation from Airlines where name="United Airlines" or country="USA"
194 union
195 select flight_number as f_no , origin as Org, destination as dest from Flights where origin="London" and destination="Paris";
196
```

Result Grid:

ID	Identity	Nation
1	American Airlines	USA
2	Delta Airlines	USA
3	United Airlines	USA
BA505	London	Paris

Action Output:

#	Time	Action	Message
1	21:43:20	select airline_id as id, name as identity, country as Nation from Airlines where name="United Airlines" or country="USA" ...	4 row(s) returned

The screenshot shows a MySQL Workbench interface with a query editor and results pane.

Query:

```
196
197 • select passenger_id as ID, passport_number as Num from passengers
198 union
199 select ticket_id as ID, seat_number from tickets;
200
```

Result Grid:

ID	Num
1020	P01232109
1010	P01234567
1011	P12345098
1001	P12345678
1012	P23450987
1002	P23456789
1013	P34509876
1003	P34567890
1014	P45608765
1004	P45678901
1015	P56707654
1005	P56789012
1016	P67806543
1006	P67890123
1007	P78901234
1017	P78905432
1018	P89004321
1008	P89012345
1019	P90103210
1009	P90123456
1	1A
2	1B

Action Output:

#	Time	Action	Message
1	21:47:10	select passenger_id as ID, passport_number as Num from passengers union select ticket_id as ID, seat_number ...	40 row(s) returned

# SUBQUERY

## a) SINGLE ROW SUBQUERY:

The screenshot shows a MySQL Workbench interface. The SQL editor window contains the following query:

```
202 •  select * from tickets where flight_id=(select flight_id from tickets where ticket_id=10);  
203  
204  
205  
206
```

The Result Grid displays the following data:

ticket_id	flight_id	passenger_id	seat_number
9	105	1009	5A
10	105	1010	5B
*	HULL	HULL	HULL

The Output pane shows the execution log:

#	Time	Action
1	22:08:53	select * from tickets where flight_id=(select flight_id from tickets where ticket_id=10)

Message: 2 row(s) returned

## b) MULTIPLE ROWS SUBQUERY:

The screenshot shows a MySQL Workbench interface. The SQL editor window contains the following query:

```
204 •  select * from passengers  
205      where passenger_id in(select passenger_id from passengers where first_name="John" or first_name="Alice");  
206  
207  
208
```

The Result Grid displays the following data:

passenger_id	first_name	last_name	passport_number
1001	Alice	Doe	P12345678
1003	Alice	Johnson	P34567890
*	HULL	HULL	HULL

The Output pane shows the execution log:

#	Time	Action
1	22:21:11	select * from passengers where passenger_id in(select passenger_id from passengers where first_name="John" ...)

Message: 2 row(s) returned

### c) MULTIPLE COLUMNS SUBQUERY:

The screenshot shows a MySQL Workbench interface. The SQL editor window contains the following code:

```
207 •  SELECT flight_id, flight_number FROM Flights WHERE (airline_id, destination)
208   IN (SELECT airline_id, 'New York' FROM Airlines WHERE country = 'USA');
209
210
211
212
```

The Result Grid shows the following output:

flight_id	flight_number
NULL	NULL

The Output pane shows the following log entry:

#	Time	Action	Message
1	22:28:34	SELECT flight_id, flight_number FROM Flights WHERE (airline_id, destination) IN (SELECT airline_id, 'New York' FROM Airlines WHERE country = 'USA');	0 row(s) returned

# JOINS

## 1. INNER JOIN:

```
211 •  select a.airline_id, name, country, flight_number, origin, destination from Airlines A
212     join Flights F
213     on a.airline_id=f.airline_id;
214
215
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

airline_id	name	country	flight_number	origin	destination
1	American Airlines	USA	AA101	New York	Los Angeles
2	Delta Airlines	USA	DL202	Los Angeles	Chicago
3	United Airlines	USA	UA303	Chicago	Houston
4	Lufthansa	Germany	LH404	Frankfurt	New York
5	British Airways	UK	BA505	London	Paris
6	Air France	France	AF606	Paris	Rome
7	Emirates	UAE	EK707	Dubai	Sydney
8	Qatar Airways	Qatar	QR808	Doha	Bangkok
9	Singapore Airlines	Singapore	SQ909	Singapore	Tokyo
10	Air Canada	Canada	AC1010	Toronto	New York
11	Qantas	Australia	QF111	Sydney	Los Angeles
12	Turkish Airlines	Turkey	TK1212	Istanbul	Berlin
13	ISIM	Malta	IS1313	Nicosia	Paris

Result 1 ×

Output:

Action Output

#	Time	Action	Message
1	11:54:46	select a.airline_id, name, country, flight_number, origin, destination from Airlines A join Flights F on a.airline_id=f.airline_id;	20 row(s) returned

## 2. LEFT JOIN:

```
215 •  select p.passenger_id, concat(first_name, ' ', last_name) as Full_name, passport_number  
216   from Passengers P  
217   left join Tickets T  
218   on p.passenger_id=t.passenger_id;  
219
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

passenger_id	Full_name	passport_number
1001	Alice Doe	P12345678
1002	Jane Smith	P23456789
1003	Alice Johnson	P34567890
1004	Bob Brown	P45678901
1005	Chris Davis	P56789012
1006	Emily Wilson	P67890123
1007	Frank Miller	P78901234
1008	Grace Taylor	P89012345
1009	Henry Anderson	P90123456
1010	Ivy Thomas	P01234567
1011	Jack Moore	P12345098
1012	Karen Martin	P23450987
1013	Leo Jackson	P34509876
1014	Mia White	P45608765
1015	Nina Harris	P56707654
1016	Oscar Clark	P67806543
1017	Paul Lewis	P78905432
1018	... " .. "	.....

Result 3 x

Output:

Action Output

#	Time	Action	Message
1	11:59:27	select p.passenger_id, concat(first_name, ' ', last_name) as Full_name, passport_number from Passengers P left join Tickets T on p.passenger_id=t.passenger_id;	20 row(s) returned

## 3. RIGHT JOIN:

```
220 •  select p.passenger_id, First_name, last_name, passport_number  
221   from Passengers P  
222   right join Tickets T  
223   on p.passenger_id=t.passenger_id;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

passenger_id	First_name	last_name	passport_number
1006	Emily	Wilson	P67890123
1007	Frank	Miller	P78901234
1008	Grace	Taylor	P89012345
1009	Henry	Anderson	P90123456
1010	Ivy	Thomas	P01234567
1011	Jack	Moore	P12345098
1012	Karen	Martin	P23450987
1013	Leo	Jackson	P34509876
1014	Mia	White	P45608765
1015	Nina	Harris	P56707654
1016	Oscar	Clark	P67806543
1017	Paul	Lewis	P78905432
1018	... " .. "	.....	.....

Result 4 x

Output:

Action Output

#	Time	Action	Message
1	12:01:16	select p.passenger_id, First_name, last_name, passport_number from Passengers P right join Tickets T on p.passenger_id=t.passenger_id;	20 row(s) returned

#### 4. FULL JOIN:

```
225 •  select a.airline_id, name, country, flight_number, origin, destination
226    from Airlines A
227    right join Flights F
228    on a.airline_id=f.airline_id
229    union
230    select a.airline_id, name, country, flight_number, origin, destination
231    from Airlines A
232    right join Flights F
233    on a.airline_id=f.airline_id;
```

Result Grid | Filter Rows: Export: Wrap Cell Content:

airline_id	name	country	flight_number	origin	destination
1	American Airlines	USA	AA101	New York	Los Angeles
2	Delta Airlines	USA	DL202	Los Angeles	Chicago
3	United Airlines	USA	UA303	Chicago	Houston
4	Lufthansa	Germany	LH404	Frankfurt	New York
5	British Airways	UK	BA505	London	Paris
6	Air France	France	AF606	Paris	Rome
7	Emirates	UAE	EK707	Dubai	Sydney
8	Qatar Airways	Qatar	QR808	Doha	Bangkok
9	Singapore Airlines	Singapore	SQ909	Singapore	Tokyo
10	Air Canada	Canada	AC1010	Toronto	New York
11	Qantas	Australia	QF111	Sydney	Los Angeles
12	Turkish Airlines	Turkey	TK1212	Istanbul	Berlin
13	KLM	Netherla...	KL1313	Amsterdam	Dubai
14	Swiss Air	Switzerland	LX1414	Zurich	London
15	Tanan Airlines	Tanan	TL1515	Tuluw	Raiiin

Result 5 ×

Output:

Action Output

#	Time	Action	Message
1	12:04:48	select a.airline_id, name, country, flight_number, origin, destination from Airlines A right join Flights F on a.airline_id...	20 row(s) returned

# VIEWS

```
236 •  create view A_view as
237      select airline_id, name, country
238      from Airlines;
```

Output:

Action Output	#	Time	Action	Message
1 12:11:06	create view A_view as select airline_id, name, country from Airlines			0 row(s) affected

```
240 •  select * from A_view;
241
242
243
244
```

airline_id	name	country
1	American Airlines	USA
2	Delta Airlines	USA
3	United Airlines	USA
4	Lufthansa	Germany
5	British Airways	UK
6	Air France	France
7	Emirates	UAE
8	Qatar Airways	Qatar
9	Singapore Airlines	Singapore
10	Air Canada	Canada
11	Qantas	Australia
12	Turkish Airlines	Turkey
13	KLM	Netherla...
14	Swiss Air	Switzerland
15	Japan Airlines	Japan
16	China Airlines	China
17	Korean Air	South Ko...
18	Aeroflot	Russia
19	Iberia	Spain
20	Alitalia	Italy

A\_view 6 ×

Output:

Action Output	#	Time	Action	Message
1 12:12:04	select * from A_view			20 row(s) returned