

HEALTH

ADMINISTRATION

DATABASES

1. INTRODUCTION

A comprehensive SQL project designed to streamline and optimize the management of hospital operations. This project aims to provide an efficient and user-friendly solution for storing, retrieving, and manipulating various types of healthcare-related data.

Healthcare administration is the management of healthcare systems, hospitals, and networks. Healthcare administrators are responsible for a wide range of tasks, including operations, finance, policies, and strategy. They need to have a combination of business, technical, hospitality, and healthcare knowledge. The primary goal is to ensure the delivery of high-quality healthcare services efficiently and effectively.

What does a healthcare administrator do?

- Analyze and manage organizational or department finances
- Assure regulatory compliance
- Facilitate meetings with senior leadership, investors and other legislative groups
- Manage facilities operations
- Recruit, hire and train new staff
- Write grant proposals to obtain research funding

Healthcare administrators need a variety of skills, including

1. Knowledge of computer programs used in healthcare offices
2. Understanding of common healthcare terminology and industry jargon
3. Strong grasp of current regulatory policies and compliance laws
4. Excellent interpersonal skills
5. Strong computer and software skills
6. Working knowledge of personnel management, record keeping, and office administration

Healthcare administrators usually have a bachelor's degree or master's degree in healthcare administration.

Project Purpose

Objectives:

1. Store and manage patient information.
2. Track appointments, bills, and insurance claims.
3. Analyze data for informed decision-making.

SQL Components:

1. Table creation (patients, appointments, bills, insurance claims).
2. Data insertion, update, and deletion.
3. Queries for data retrieval and analysis.
4. Indexing and constraints for data integrity.



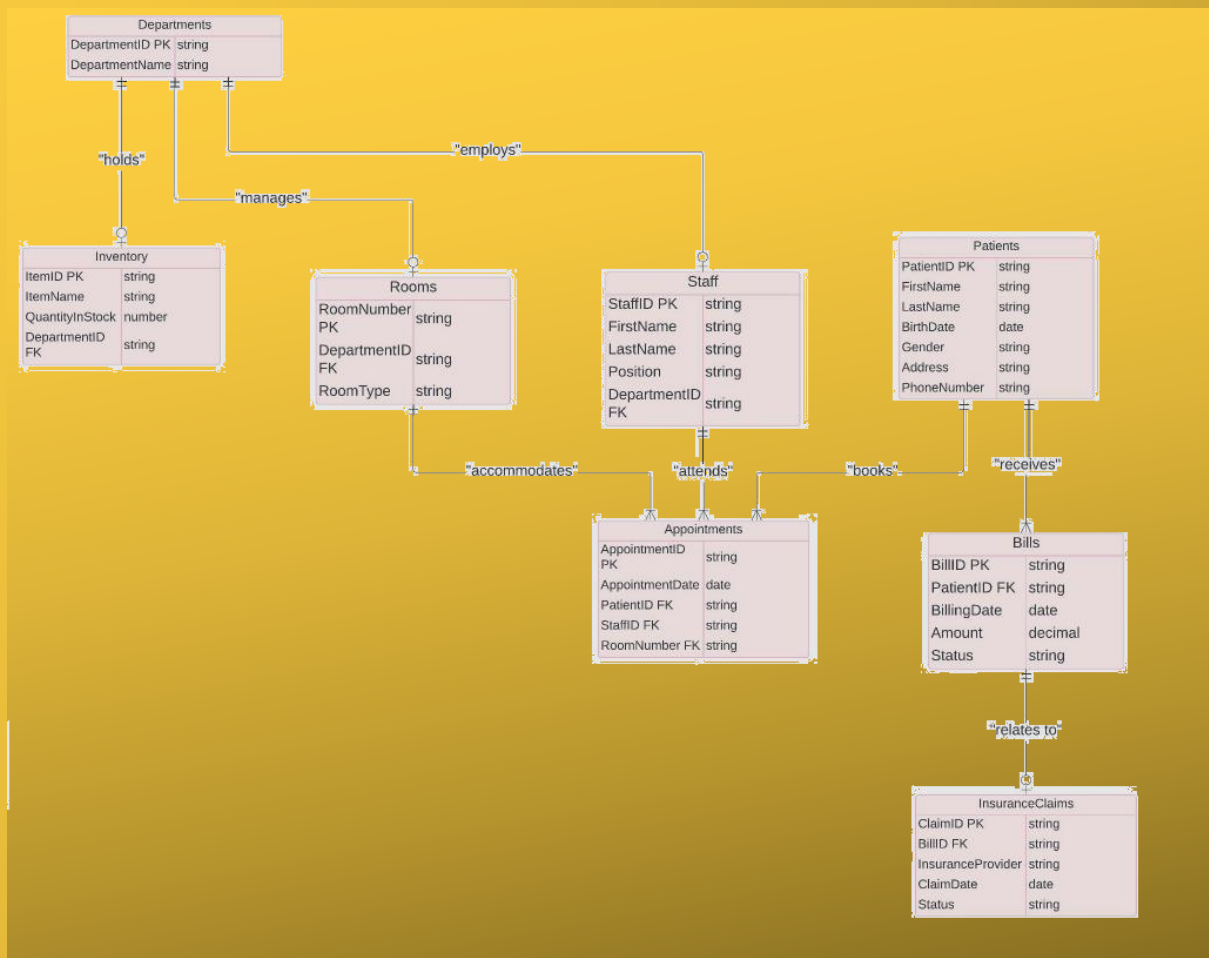
2.DATABASE DESIGN

DATABASES -: HEALTH ADMINISTRATION

TABLES -:

- **Patients**
- **Appointments**
- **Staff**
- **Bills**
- **Insurance Claims**
- **Inventory**
- **DEPARTMENTS**
- **ROOMS**

3.ER DIAGRAM



4.Creating Table

A) Patients

```
CREATE TABLE Patients (  
  PatientID VARCHAR(255) PRIMARY KEY,  
  FirstName VARCHAR(255) NOT NULL,  
  LastName VARCHAR(255) NOT NULL,  
  BirthDate DATE NOT NULL,  
  Gender VARCHAR(255) NOT NULL,  
  Address VARCHAR(255),  
  PhoneNumber VARCHAR(255)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

PatientID	FirstName	LastName	BirthDate	Gender	Address	PhoneNumber
-----------	-----------	----------	-----------	--------	---------	-------------

empty

B) Appointments

```
CREATE TABLE Appointments (  
  AppointmentID VARCHAR(255) PRIMARY KEY,  
  AppointmentDate DATE NOT NULL,  
  PatientID VARCHAR(255) NOT NULL,  
  StaffID VARCHAR(255) NOT NULL,  
  RoomNumber VARCHAR(255) NOT NULL,  
  FOREIGN KEY (PatientID) REFERENCES Patients(PatientID),  
  FOREIGN KEY (StaffID) REFERENCES Staff(StaffID),  
  FOREIGN KEY (RoomNumber) REFERENCES Rooms(RoomNumber)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

AppointmentID	AppointmentDate	PatientID	StaffID	RoomNumber
---------------	-----------------	-----------	---------	------------

empty

C) Staff

```
CREATE TABLE Staff (  
  StaffID VARCHAR(255) PRIMARY KEY,  
  FirstName VARCHAR(255) NOT NULL,  
  LastName VARCHAR(255) NOT NULL,  
  Position VARCHAR(255) NOT NULL,  
  DepartmentID VARCHAR(255) NOT NULL,  
  FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

StaffID	FirstName	LastName	Position	DepartmentID
---------	-----------	----------	----------	--------------

empty

D) Bills

```
CREATE TABLE Bills (  
  BillID VARCHAR(255) PRIMARY KEY,  
  PatientID VARCHAR(255) NOT NULL,  
  BillingDate DATE NOT NULL,  
  Amount DECIMAL(10, 2) NOT NULL,  
  Status VARCHAR(255) NOT NULL CHECK (Status IN ('Paid', 'Pending', 'Overdue')),  
  FOREIGN KEY (PatientID) REFERENCES Patients(PatientID)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

BillID	PatientID	BillingDate	Amount	Status
--------	-----------	-------------	--------	--------

empty

E) Insurance Claims

```
CREATE TABLE InsuranceClaims (  
  ClaimID VARCHAR(255) PRIMARY KEY,  
  BillID VARCHAR(255) NOT NULL,  
  InsuranceProvider VARCHAR(255) NOT NULL,  
  ClaimDate DATE NOT NULL,  
  Status VARCHAR(255) NOT NULL CHECK (Status IN ('Pending', 'Approved', 'Denied',  
'Paid')),  
  FOREIGN KEY (BillID) REFERENCES Bills(BillID)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

ClaimID	BillID	InsuranceProvider	ClaimDate	Status
empty				

F) Inventory

```
CREATE TABLE Inventory (  
  ItemID VARCHAR(255) PRIMARY KEY,  
  ItemName VARCHAR(255) NOT NULL,  
  QuantityInStock INT NOT NULL DEFAULT 0,  
  DepartmentID VARCHAR(255) NOT NULL,  
  FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

ItemID	ItemName	QuantityInStock	DepartmentID
empty			

G) Departments


```
CREATE TABLE Departments (  
    DepartmentID VARCHAR(255) PRIMARY KEY,  
    DepartmentName VARCHAR(255) NOT NULL UNIQUE  
);
```

Output

SQL query successfully executed. However, the result set is empty.

DepartmentID	DepartmentName
empty	

H) Rooms

```
CREATE TABLE Rooms (  
    RoomNumber VARCHAR(255) PRIMARY KEY,  
    DepartmentID VARCHAR(255) NOT NULL,  
    RoomType VARCHAR(255) NOT NULL,  
    FOREIGN KEY (DepartmentID) REFERENCES Departments(DepartmentID)  
);
```

Output

SQL query successfully executed. However, the result set is empty.

RoomNumber	DepartmentID	RoomType
empty		

5.Tables in Databases

688 • SHOW TABLES;

689

690

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

Tables_in_rr

appointments

bills

departments

insuranceclaims

inventory

patients

Result 5 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
53	00:04:45	CREATE TABLE Bills (BillID VARCHAR(255) PRIMARY KEY, PatientID VARCHAR(255) NOT NULL, BillingDate DATE NOT ...	0 row(s) affected	0.047 sec
54	00:04:49	CREATE TABLE InsuranceClaims (ClaimID VARCHAR(255) PRIMARY KEY, BillID VARCHAR(255) NOT NULL, InsurancePr...	0 row(s) affected	0.047 sec
55	00:05:04	SHOW TABLES	8 row(s) returned	0.000 sec / 0.000 sec

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

Tables_in_rr

appointments

bills

departments

insuranceclaims

inventory

patients

rooms

staff

Result 5 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
53	00:04:45	CREATE TABLE Bills (BillID VARCHAR(255) PRIMARY KEY, PatientID VARCHAR(255) NOT NULL, BillingDate DATE NOT ...	0 row(s) affected	0.047 sec
54	00:04:49	CREATE TABLE InsuranceClaims (ClaimID VARCHAR(255) PRIMARY KEY, BillID VARCHAR(255) NOT NULL, InsurancePr...	0 row(s) affected	0.047 sec
55	00:05:04	SHOW TABLES	8 row(s) returned	0.000 sec / 0.000 sec

6. Data Definition language (DDL)

A) Creating Tables:

1. Patients

690 • desc patients;

691

Field	Type	Null	Key	Default	Extra
PatientID	varchar(255)	NO	PRI	NULL	
FirstName	varchar(255)	NO		NULL	
LastName	varchar(255)	NO		NULL	
BirthDate	date	NO		NULL	
Gender	varchar(255)	NO		NULL	
Address	varchar(255)	YES		NULL	
PhoneNumber	varchar(255)	YES		NULL	

Result 6 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
55	00:05:04	SHOW TABLES	8 row(s) returned	0.000 sec / 0.000 sec
56	00:13:06	desc patients	7 row(s) returned	0.016 sec / 0.000 sec

2. Appointments

691 • desc appointments;

692

Field	Type	Null	Key	Default	Extra
AppointmentID	varchar(255)	NO	PRI	NULL	
AppointmentDate	date	NO		NULL	
PatientID	varchar(255)	NO	MUL	NULL	
StaffID	varchar(255)	NO	MUL	NULL	
RoomNumber	varchar(255)	NO	MUL	NULL	

Result 7 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
56	00:13:06	desc patients	7 row(s) returned	0.016 sec / 0.000 sec
57	00:16:35	desc appointments	5 row(s) returned	0.000 sec / 0.000 sec

3. Staff

692 • desc staff;

693

694

695

696

697

698

699

700

toggle automatic help.

Result Grid

Field	Type	Null	Key	Default	Extra
StaffID	varchar(255)	NO	PRI	NULL	
FirstName	varchar(255)	NO		NULL	
LastName	varchar(255)	NO		NULL	
Position	varchar(255)	NO		NULL	
DepartmentID	varchar(255)	NO	MUL	NULL	

Result 8 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
57	00:16:35	desc appointments	5 row(s) returned	0.000 sec / 0.000 sec
58	00:18:24	desc staff	5 row(s) returned	0.016 sec / 0.000 sec

4.Bills

694 • desc bills;

695

696

697

698

699

700

Result Grid

Field	Type	Null	Key	Default	Extra
BillID	varchar(255)	NO	PRI	NULL	
PatientID	varchar(255)	NO	MUL	NULL	
BillingDate	date	NO		NULL	
Amount	decimal(10,2)	NO		NULL	
Status	varchar(255)	NO		NULL	

Result 10 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
60	01:13:44	desc bills	Error Code: 1146. Table 'bills' doesn't exist	0.000 sec
61	01:13:53	desc bills	5 row(s) returned	0.015 sec / 0.000 sec

5. Insurance Claims

695 desc InsuranceClaims;
696

Result Grid

Filter Rows:

Export:

Wrap Cell Contents:

Field	Type	Null	Key	Default	Extra
ClaimID	varchar(255)	NO	PR1	NULL	
BillID	varchar(255)	NO	MUL	NULL	
InsuranceProvider	varchar(255)	NO		NULL	
ClaimDate	date	NO		NULL	
Status	varchar(255)	NO		NULL	

Result 11 x

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / F
62	01:20:29	desc insurance claims	Error Code: 1146. Table 'insurance' doesn't exist	0.000 sec
63	01:21:34	desc InsuranceClaims	5 row(s) returned	0.015 sec /

6. Inventory

696 • desc Inventory;

697

Result Grid Filter Rows: Export: Wrap Cell Contents:

Field	Type	Null	Key	Default	Extra
ItemID	varchar(255)	NO	PRI	NULL	
ItemName	varchar(255)	NO		NULL	
QuantityInStock	int	NO		0	
DepartmentID	varchar(255)	NO	MUL	NULL	

Result 12 x [Context Help](#) [Snippets](#)

Output

Action Output

#	Time	Action	Message	Duration / Fetch
63	01:21:34	desc InsuranceClaims	5 row(s) returned	0.015 sec / 0.000 sec
64	01:23:34	desc inventory	4 row(s) returned	0.016 sec / 0.000 sec

7. Departments

693 • desc departments;

694

695

696

697

698

699

700

Result Grid Filter Rows: Export: Wrap Cell Contents:

Field	Type	Null	Key	Default	Extra
DepartmentID	varchar(255)	NO	PRI	NULL	
DepartmentName	varchar(255)	NO	UNI	NULL	

Result 9 x [Context Help](#) [Snippets](#)

Output

Action Output

#	Time	Action	Message	Duration / Fetch
58	00:18:24	desc staff	5 row(s) returned	0.016 sec / 0.000 sec
59	00:19:41	desc departments	2 row(s) returned	0.000 sec / 0.000 sec

8.Rooms

697 • desc rooms

698

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents: |

Field	Type	Null	Key	Default	Extra
RoomNumber	varchar(255)	NO	PRI	NULL	
DepartmentID	varchar(255)	NO	MUL	NULL	
RoomType	varchar(255)	NO		NULL	

Result 13 x

Read Only Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
64	01:23:34	desc inventory	4 row(s) returned	0.016 sec / 0.000 sec
65	01:26:38	desc rooms	3 row(s) returned	0.000 sec / 0.000 sec

B)Alter table

1) Alter table add column:

698 • alter table patients add

699 dateofbirth date;

700 • ALTER TABLE Patients

701 ADD EmergencyContactName VARCHAR(100);

702

703

704

705

706

707

708

709

710

711

712

Context Help Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
67	01:40:39	alter table patients add dateofbirth date	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.063 sec
68	01:42:16	ALTER TABLE Patients ADD EmergencyContactName VARCHAR(100)	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.047 sec

Describe Patients

Result Grid | Filter Rows: | Exports: | Wrap Cell Contents: |

Field	Type	Null	Key	Default	Extra
PatientID	varchar(255)	NO	PRI	NULL	
FirstName	varchar(255)	NO		NULL	
LastName	varchar(255)	NO		NULL	
BirthDate	date	NO		NULL	
Gender	varchar(255)	NO		NULL	
Address	varchar(255)	YES		NULL	
PhoneNumber	varchar(255)	YES		NULL	
dateofbirth	date	YES		NULL	
EmergencyContactName	varchar(100)	YES		NULL	

Result 16 x

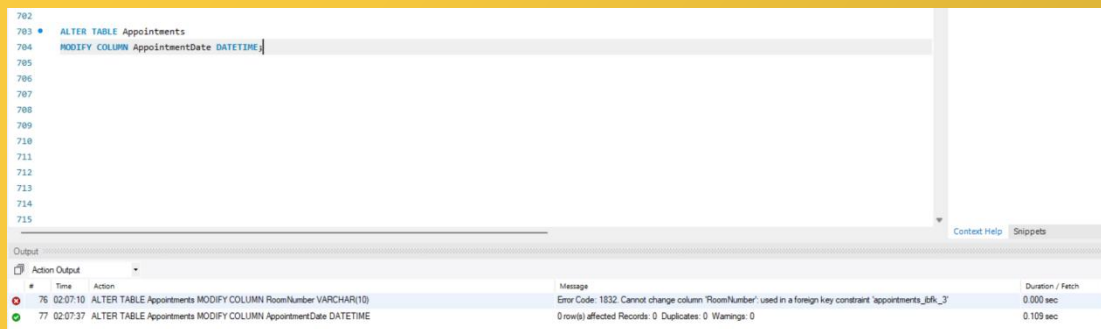
Read Only Context Help Snippets

Output

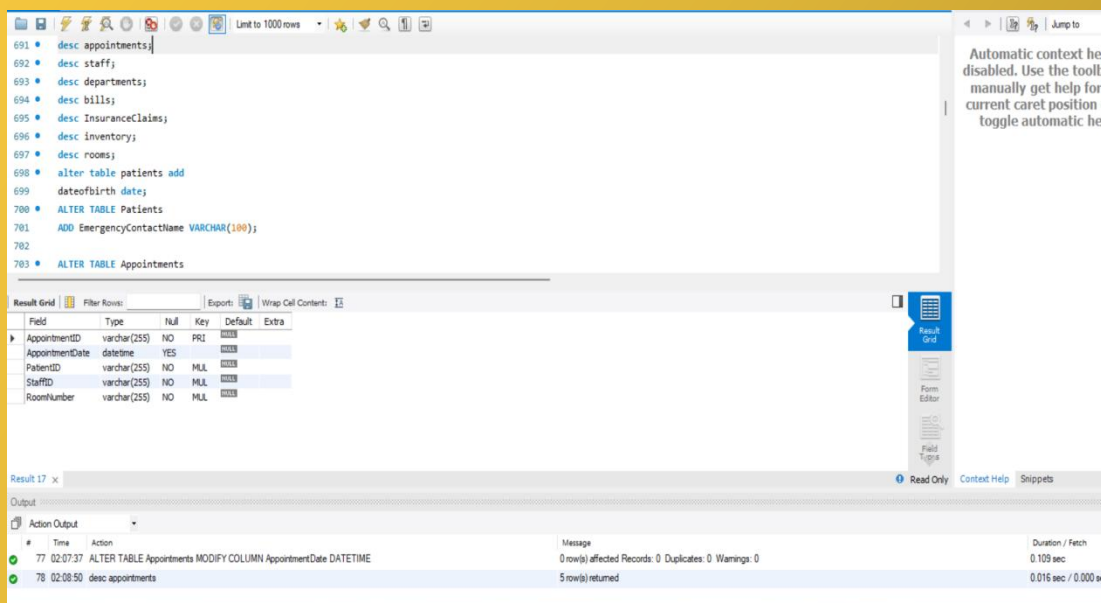
Action Output

#	Time	Action	Message	Duration / Fetch
70	01:45:10	select * from patients LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
71	01:45:33	desc patients	9 row(s) returned	0.000 sec / 0.000 sec

2) Alter table modify column:



Describe Appointments



3) Alter table rename column:

```

706 • ALTER TABLE Patients
707     RENAME COLUMN FirstName TO PatientFirstName,
708     RENAME COLUMN LastName TO PatientLastName;
709
710
711
712
713
714
715
716
717
718
719
720
721

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
79	02:16:47	ALTER TABLE Patients RENAME COLUMN FirstName TO PatientFirstName, RENAME COLUMN LastName TO PatientLastNam...	Error Code: 1060. Duplicate column name 'dateofbirth'	0.000 sec
80	02:17:08	ALTER TABLE Patients RENAME COLUMN FirstName TO PatientFirstName, RENAME COLUMN LastName TO PatientLastName	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Describe Patients

```

709 • desc patients;
710

```

Result Grid

Field	Type	Null	Key	Default	Extra
PatientID	varchar(255)	NO	PRI	0000	
PatientFirstName	varchar(255)	NO		0000	
PatientLastName	varchar(255)	NO		0000	
BirthDate	date	NO		0000	
Gender	varchar(255)	NO		0000	
Address	varchar(255)	YES		0000	
PhoneNumber	varchar(255)	YES		0000	
dateofbirth	date	YES		0000	
EmergencyContactName	varchar(100)	YES		0000	

Output

Action Output

#	Time	Action	Message	Duration / Fetch
80	02:17:08	ALTER TABLE Patients RENAME COLUMN FirstName TO PatientFirstName, RENAME COLUMN LastName TO PatientLastName	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
81	02:19:35	desc patients	9 row(s) returned	0.000 sec / 0.000 sec

4) Alter table drop column:

```

711 • ALTER TABLE patients
712     DROP COLUMN birthdate;
713
714
715
716
717
718
719
720
721
722
723
724

```

Output

Action Output

#	Time	Action	Message	Duration / Fetch
81	02:19:35	desc patients	9 row(s) returned	0.000 sec / 0.000 sec
82	02:23:51	ALTER TABLE patients DROP COLUMN birthdate	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec

Describe Patients

```

709 • desc patients;
710
711 • ALTER TABLE patients
712   DROP COLUMN birthdate;
713

```

Field	Type	Null	Key	Default	Extra
PatientID	varchar(255)	NO	PRI	NULL	
PatientFirstName	varchar(255)	NO		NULL	
PatientLastName	varchar(255)	NO		NULL	
Gender	varchar(255)	NO		NULL	
Address	varchar(255)	YES		NULL	
PhoneNumber	varchar(255)	YES		NULL	
dateofBirth	date	YES		NULL	
EmergencyContactName	varchar(100)	YES		NULL	

#	Time	Action	Message	Duration / Fetch
81	02:19:35	desc patients	9 row(s) returned	0.000 sec / 0.000 sec
82	02:23:51	ALTER TABLE patients DROP COLUMN birthdate	0 row(s) affected Records: 0 Duplicates: 0 Warnings: 0	0.031 sec
83	02:24:42	desc patients	8 row(s) returned	0.015 sec / 0.000 sec

C) Rename table

```

714 • RENAME TABLE Patients TO PatientInformation;
715
716
717
718
719
720
721
722
723
724
725
726

```

#	Time	Action	Message	Duration / Fetch
83	02:24:42	desc patients	8 row(s) returned	0.015 sec / 0.000 sec
84	02:29:45	RENAME TABLE Patients TO PatientInformation	0 row(s) affected	0.031 sec

Describe PatientInformation

```

716 • desc PatientInformation;
717
718

```

Field	Type	Null	Key	Default	Extra
PatientID	varchar(255)	NO	PRI	NULL	
PatientFirstName	varchar(255)	NO		NULL	
PatientLastName	varchar(255)	NO		NULL	
Gender	varchar(255)	NO		NULL	
Address	varchar(255)	YES		NULL	
PhoneNumber	varchar(255)	YES		NULL	
dateofBirth	date	YES		NULL	
EmergencyContactName	varchar(100)	YES		NULL	

#	Time	Action	Message	Duration / Fetch
84	02:29:45	RENAME TABLE Patients TO PatientInformation	0 row(s) affected	0.031 sec
85	02:31:30	desc PatientInformation	8 row(s) returned	0.000 sec / 0.000 sec

D) Truncate table

Create Truncate_practice table


```
717
718 • CREATE TABLE truncate_practice (
719     id INT PRIMARY KEY,
720     name VARCHAR(255),
721     email VARCHAR(255),
722     age INT
723 );
724
725 • desc truncate_practice;
```

Output

#	Time	Action	Message	Duration / Fetch
88	02:44:45	TRUNCATE TABLE test	Error Code: 1146: Table 'test' doesn't exist	0.000 sec
89	02:46:42	CREATE TABLE truncate_practice (id INT PRIMARY KEY, name VARCHAR(255), email VARCHAR(255), age INT)	0 row(s) affected	0.046 sec

Describe Truncate_practice

```
724
725 • desc truncate_practice;
```

Result Grid

Field	Type	Null	Key	Default	Extra
id	int	NO	PR		
name	varchar(255)	YES			
email	varchar(255)	YES			
age	int	YES			

Output

#	Time	Action	Message	Duration / Fetch
89	02:46:42	CREATE TABLE truncate_practice (id INT PRIMARY KEY, name VARCHAR(255), email VARCHAR(255), age INT)	0 row(s) affected	0.046 sec
90	02:48:22	desc truncate_practice	4 row(s) returned	0.000 sec / 0.000 sec

TRUNCATE TABLE truncate_practice

```
726 • TRUNCATE TABLE truncate_practice;
```

Output

#	Time	Action	Message	Duration / Fetch
90	02:48:22	desc truncate_practice	4 row(s) returned	0.000 sec / 0.000 sec
91	02:51:15	TRUNCATE TABLE truncate_practice	0 row(s) affected	0.047 sec

SELECT * FROM truncate_practice

726 TRUNCATE TABLE truncate_practice;
727 SELECT * FROM truncate_practice;
728

Result Grid

#	id	name	email	age
1	1000	1000	1000	1000

truncate_practice 22 x

Output

#	Time	Action	Message	Duration / Fetch
91	02:51:15	TRUNCATE TABLE truncate_practice	0 row(s) affected	0.047 sec
92	02:52:14	SELECT * FROM truncate_practice LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec

E) Drop table

728
729 DROP TABLE truncate_practice;
730
731
732
733
734
735
736
737
738
739
740
741

Output

#	Time	Action	Message	Duration / Fetch
92	02:52:14	SELECT * FROM truncate_practice LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
93	03:02:00	DROP TABLE truncate_practice	0 row(s) affected	0.031 sec

7. Data Manipulation language (DML)

A) Insert into table

731
732 INSERT INTO PatientInformation (PatientID, PatientFirstName, PatientLastName, Gender, Address, PhoneNumber, DateOfBirth, EmergencyContactName)
733 VALUES ('PAT001', 'John', 'Doe', 'Male', '123 Main St', '123-456-7890', '1990-01-01', 'Jane Doe'),
734 ('PAT002', 'Jane', 'Smith', 'Female', '456 Elm St', '987-654-3210', '1995-06-01', 'Bob Smith');
735
736
737
738
739
740
741
742
743
744

Output

#	Time	Action	Message	Duration / Fetch
106	03:16:20	desc PatientInformation	8 row(s) returned	0.000 sec / 0.000 sec
107	03:22:29	INSERT INTO PatientInformation (PatientID, PatientFirstName, PatientLastName, Gender, Address, PhoneNumber, DateOfBirth, Emer...	2 row(s) affected Records: 2 Duplicates: 0 Warnings: 0	0.016 sec

B) Update into table

```

737 • UPDATE PatientInformation
738 SET PhoneNumber = '999-888-7777', Address = '789 New St'
739 WHERE PatientID = 'PAT001';
740
741
742
743
744
745
746
747
748
749
750
751

```

Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
109	13:05:19	UPDATE Patients SET PhoneNumber = '999-888-7777', Address = '789 New St' WHERE PatientID = 'PAT001'	Error Code: 1146, Table 'r.patients' doesn't exist	0.000 sec
110	13:05:46	UPDATE PatientInformation SET PhoneNumber = '999-888-7777', Address = '789 New St' WHERE PatientID = 'PAT001'	1 row(s) affected Rows matched: 1 Changed: 1 Warnings: 0	0.016 sec

C) Delete into table

```

743 • DELETE FROM PatientInformation
744 WHERE PatientID = 'PAT001';
745
746
747
748
749
750
751
752
753
754
755
756

```

Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
111	13:07:19	SELECT * FROM PatientInformation WHERE PatientID = 'PAT001' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
112	13:11:10	DELETE FROM PatientInformation WHERE PatientID = 'PAT001'	1 row(s) affected	0.016 sec

8. Data Query Language (DQL)

A) Select query

```

745
746 • SELECT * FROM PatientInformation
747
748

```

Result Grid

PatientID	PatientFirstName	PatientLastName	Gender	Address	PhoneNumber	dateofbirth	EmergencyContactName
PAT002	Jane	Smith	Female	456 Elm St	987-654-3210	1995-06-01	Bob Smith

Apply Context Help Snippets

Output

#	Time	Action	Message	Duration / Fetch
112	13:11:10	DELETE FROM PatientInformation WHERE PatientID = 'PAT001'	1 row(s) affected	0.016 sec
113	13:12:19	SELECT * FROM PatientInformation LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

Find all staff

Automatic context help is disabled. Use the toolbar to manually get help for the current caret position or to toggle automatic help.

```

650 • INSERT INTO Staff (StaffID, FirstName, LastName, Position, DepartmentID)
651 VALUES
652 ('STF001', 'John', 'Doe', 'Doctor', 'DEPT001'),
653 ('STF002', 'Jane', 'Smith', 'Nurse', 'DEPT002'),
654 ('STF003', 'Bob', 'Johnson', 'Admin', 'DEPT003');
655 • SELECT * FROM Staff;
656
657 • CREATE TABLE Departments (
658     DepartmentID VARCHAR(255) PRIMARY KEY,
659     DepartmentName VARCHAR(255) NOT NULL UNIQUE
660 );
661 • INSERT INTO Departments (DepartmentID, DepartmentName)

```

StaffID	FirstName	LastName	Position	DepartmentID
STF001	John	Doe	Doctor	DEPT001
STF002	Jane	Smith	Nurse	DEPT002
STF003	Bob	Johnson	Admin	DEPT003

Staff 30 x

Output

#	Time	Action	Message	Duration / Fetch
126	13:44:53	INSERT INTO Staff (StaffID, FirstName, LastName, Position, DepartmentID) VALUES ('STF001', 'John', 'Doe', 'Doctor', 'DEPT001'), ('STF002', 'Jane', 'Smith', 'Nurse', 'DEPT002'), ('STF003', 'Bob', 'Johnson', 'Admin', 'DEPT003');	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.015 sec
127	13:45:26	SELECT * FROM Staff LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec

B) Order by query ASC.

"How can you display all departments sorted by their Department IDs in ascending order?"

manually get help for the current caret position or to toggle automatic help.

```

767 /*=====Order by DepartmentID ASC=====*/
768
769 • SELECT *
770 FROM Departments
771 ORDER BY DepartmentID ASC;
772
773
774
775
776

```

DepartmentID	DepartmentName
DEPT001	Cardiology
DEPT002	Neurology
DEPT003	Administration

Departments 31 x

Output

#	Time	Action	Message	Duration / Fetch
127	13:45:26	SELECT * FROM Staff LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec
128	14:33:15	SELECT * FROM Departments ORDER BY DepartmentID ASC LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

C) Order by query DESC

"How can you display all departments sorted by their Department IDs in descending order?"

776

777 • `SELECT FirstName, LastName, Position`

778 `FROM Staff`

779 `ORDER BY LastName DESC;`

780

Result Grid

FirstName	LastName	Position
Jane	Smith	Nurse
Bob	Johnson	Admin
John	Doe	Doctor

Staff 35 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
131	14:40:23	<code>SELECT * FROM Staff ORDER BY FirstName DESC LIMIT 0, 1000</code>	3 row(s) returned	0.000 sec / 0.000 sec
132	14:42:39	<code>SELECT FirstName, LastName, Position FROM Staff ORDER BY LastName DESC LIMIT 0, 1000</code>	3 row(s) returned	0.000 sec / 0.000 sec

D) Order by column

"Show all items from the inventory with their details, sorted from the largest to the smallest quantity in stock."

781 • `SELECT *`

782 `FROM Inventory`

783 `ORDER BY QuantityInStock DESC;`

784

Result Grid

ItemID	ItemName	QuantityInStock	DepartmentID
ITM003	Medication	200	DEPT003
ITM001	Bandages	100	DEPT001
ITM002	Syringes	50	DEPT002

Inventory 15 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
23	15:10:45	<code>SELECT * FROM Inventory ORDER BY DepartmentID ASC LIMIT 0, 1000</code>	3 row(s) returned	0.000 sec / 0.000 sec
24	15:11:24	<code>SELECT * FROM Inventory ORDER BY DepartmentID desc, ItemName desc LIMIT 0, 1000</code>	3 row(s) returned	0.000 sec / 0.000 sec
25	15:11:52	<code>SELECT * FROM Inventory ORDER BY QuantityInStock DESC LIMIT 0, 1000</code>	3 row(s) returned	0.000 sec / 0.000 sec

E) Limit query

Write a SQL query to retrieve only the first 2 rows from the Bills table

735 • `SELECT * FROM bills;`

736 • `select * from bills limit 2;`

737

738 • `select Patients`

739 `from SQLbatch4to9 tables;`

740

Result Grid

BillID	PatientID	BillingDate	Amount	Status
BILL001	PAT001	2024-10-21	100.00	Pending
BILL002	PAT002	2024-10-22	200.00	Paid

bills 29 x

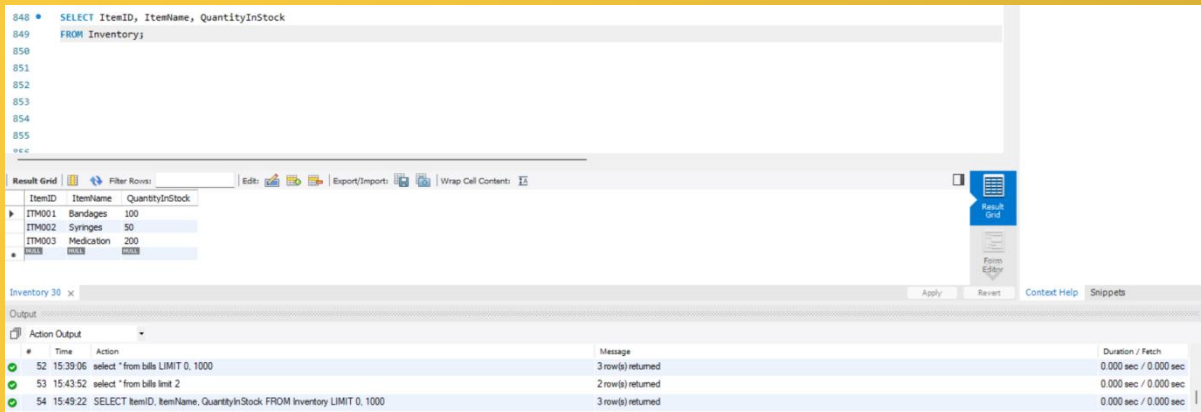
Output

Action Output

#	Time	Action	Message	Duration / Fetch
51	15:39:03	<code>INSERT INTO Bills (BillID, PatientID, BillingDate, Amount, Status) VALUES ('BILL001', 'PAT001', '2024-10-21', 100.00, 'Pending');</code>	3 row(s) affected Records: 3 Duplicates: 0 Warnings: 0	0.000 sec
52	15:39:06	<code>select * from bills LIMIT 0, 1000</code>	3 row(s) returned	0.000 sec / 0.000 sec
53	15:43:52	<code>select * from bills limit 2</code>	2 row(s) returned	0.000 sec / 0.000 sec

F) Select query with specific column

To retrieve the ItemID, ItemName, and QuantityInStock from the Inventory table.



```
848 • SELECT ItemID, ItemName, QuantityInStock
849 FROM Inventory;
850
851
852
853
854
855
856
```

ItemID	ItemName	QuantityInStock
ITM001	Bandages	100
ITM002	Syringes	50
ITM003	Medication	200

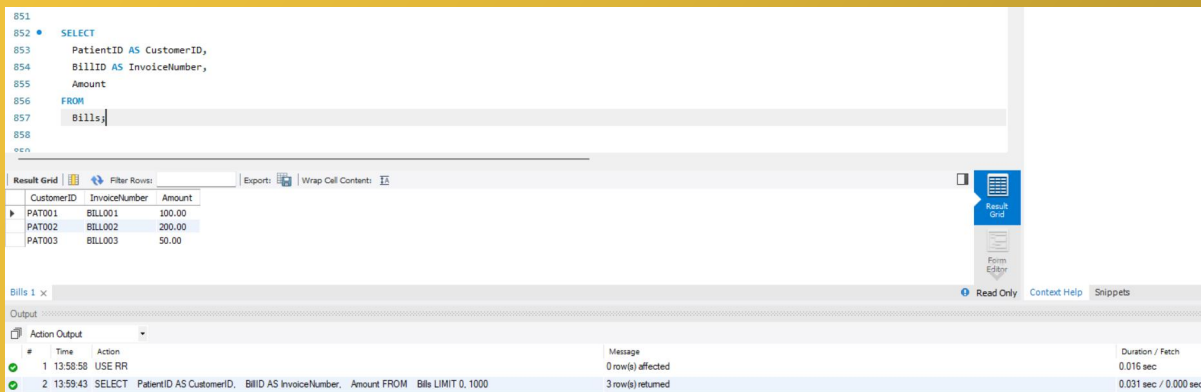
Inventory 30 x

Output

#	Time	Action	Message	Duration / Fetch
52	15:39:06	select * from bills LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec
53	15:43:52	select * from bills limit 2	2 row(s) returned	0.000 sec / 0.000 sec
54	15:49:22	SELECT ItemID, ItemName, QuantityInStock FROM Inventory LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

G) Select query with column name change

Retrieve patient ID as Customer ID ,Bill ID as Invoice Number and Amount From Bills Table



```
851
852 • SELECT
853     PatientID AS CustomerID,
854     BillID AS InvoiceNumber,
855     Amount
856 FROM
857     Bills;
858
859
```

CustomerID	InvoiceNumber	Amount
PAT001	BILL001	100.00
PAT002	BILL002	200.00
PAT003	BILL003	50.00

Bills 1 x

Output

#	Time	Action	Message	Duration / Fetch
1	13:58:58	USE RR	0 row(s) affected	0.016 sec
2	13:59:43	SELECT PatientID AS CustomerID, BillID AS InvoiceNumber, Amount FROM Bills LIMIT 0, 1000	3 row(s) returned	0.031 sec / 0.000 sec

H) Distinct query

Retrieve unique Item Names from Inventory table

858

859 • SELECT DISTINCT ItemName

860 FROM Inventory

861

Result Grid

Filter Rows:

Export

Wrap Cell Contents

ItemName

Bandages

Syringes

Medication

Inventory 2 x

Read Only

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
1	13:58:58	USE RR	0 row(s) affected	0.016 sec
2	13:59:43	SELECT PatientID AS CustomerID, BILLID AS InvoiceNumber, Amount FROM Bills LIMIT 0, 1000	3 row(s) returned	0.031 sec / 0.000 sec
3	14:12:20	SELECT DISTINCT ItemName FROM Inventory LIMIT 0, 1000	3 row(s) returned	0.016 sec / 0.000 sec

9. Using Where clause

A) With Comparision Operator

Retrieve items with QuantityInStock < 200

862

863 • SELECT *

864 FROM Inventory

865 WHERE QuantityInStock < 200

866

867

Result Grid

Filter Rows

Edit

Export/Import

Wrap Cell Contents

Inventory 7 x

Output

Action Output

#	Time	Action	Message	Duration / Fetch
7	14:18:17	SELECT * FROM Inventory WHERE QuantityInStock < 50 LIMIT 0, 1000	0 row(s) returned	0.000 sec / 0.000 sec
8	14:19:28	CREATE TABLE Inventory (ItemID VARCHAR(255) PRIMARY KEY, ItemName VARCHAR(255) NOT NULL, QuantityInStock...	Error Code: 1050. Table 'Inventory' already exists	0.000 sec
9	14:20:32	SELECT * FROM Inventory WHERE QuantityInStock < 200 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec

ItemID	ItemName	QuantityInStock	DepartmentID
ITM001	Bandages	100	DEPT001
ITM002	Syringes	50	DEPT002

Result Grid

Form Editor

Apply

Context Help

Snippets

Retrieve bills with Status = 'Paid'

866

867 • SELECT *

868 FROM Bills

869 WHERE Status = 'Paid';

870

Result Grid

Filter Rows

Edit

Export/Import

Wrap Cell Contents

15

BILLID	PatientID	BillingDate	Amount	Status
BILL002	PAT002	2024-10-22	200.00	Paid

Result Grid

Form Editor

Bills 8 x

Apply

Context Help

Snippets

Output

Action Output

#	Time	Action	Message	Duration / Fetch
8	14:19:28	CREATE TABLE Inventory (ItemID VARCHAR(255) PRIMARY KEY, ItemName VARCHAR(255) NOT NULL, QuantityInStoc...	Error Code: 1050. Table 'Inventory' already exists	0.000 sec
9	14:20:32	SELECT * FROM Inventory WHERE QuantityInStock < 200 LIMIT 0, 1000	2 row(s) returned	0.000 sec / 0.000 sec
10	14:26:25	SELECT * FROM Bills WHERE Status = 'Paid' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

10. Using Logical Operator

A) Using AND operator

Retrieve bills with Amount > 100 and Status = 'Paid'

The screenshot shows a database query editor with the following SQL query:

```
738 SELECT *  
739 FROM Bills  
740 WHERE Amount > 100 AND Status = 'Paid';  
741
```

The query is executed, and the results are displayed in a table:

BILLID	PatientID	BillingDate	Amount	Status
BILL002	PAT002	2024-10-22	200.00	Paid

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
11	14:30:49	SELECT * FROM PatientInformation WHERE Age > 60 AND Gender = 'Male' LIMIT 0, 1000	Error Code: 1054. Unknown column 'Age' in 'where clause'	0.016 sec
12	14:33:36	SELECT * FROM Bills WHERE Amount > 100 AND PaymentStatus = 'Paid' LIMIT 0, 1000	Error Code: 1054. Unknown column 'PaymentStatus' in 'where clause'	0.000 sec
13	14:35:13	SELECT * FROM Bills WHERE Amount > 100 AND Status = 'Paid' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

B) Using AND/ OR operator

To retrieve staff members working in Emergency and holding the Nurse position, use OR

The screenshot shows a database query editor with the following SQL query:

```
878 SELECT *  
879 FROM Staff  
880 WHERE DepartmentID = 'Emergency' OR Position = 'Nurse';  
881
```

The query is executed, and the results are displayed in a table:

StaffID	FirstName	LastName	Position	DepartmentID
STF002	Jane	Smith	Nurse	DEPT002

The output pane shows the execution log:

#	Time	Action	Message	Duration / Fetch
13	14:35:13	SELECT * FROM Bills WHERE Amount > 100 AND Status = 'Paid' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
14	14:45:22	SELECT * FROM Staff WHERE Department = 'Emergency' OR Position = 'Nurse' LIMIT 0, 1000	Error Code: 1054. Unknown column 'Department' in 'where clause'	0.016 sec
15	14:46:34	SELECT * FROM Staff WHERE DepartmentID = 'Emergency' OR Position = 'Nurse' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

C) Using between clause

Retrieve bills with Amount between 50 And 300

```
885 • SELECT *
886 FROM Bills
887 WHERE Amount BETWEEN 50 AND 300
888
889
890
891
892
```

BILLID	PatientID	BillingDate	Amount	Status
BILL001	PAT001	2024-10-21	100.00	Pending
BILL002	PAT002	2024-10-22	200.00	Paid
BILL003	PAT003	2024-10-23	50.00	Overdue

Output

#	Time	Action	Message	Duration / Fetch
14	14:45:22	SELECT * FROM Staff WHERE Department = 'Emergency' OR Position = 'Nurse' LIMIT 0, 1000	Error Code: 1054. Unknown column 'Department' in 'where clause'	0.016 sec
15	14:46:34	SELECT * FROM Staff WHERE DepartmentID = 'Emergency' OR Position = 'Nurse' LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
16	14:54:00	SELECT * FROM Bills WHERE Amount BETWEEN 50 AND 300 LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

D) Using IN clause

Retrieve details of patients with IDs “PAT002”

```
861 • SELECT *
862 FROM PatientInformation
863 WHERE PatientID IN ( 'PAT002' );
864
865
866
867
868
869
870
871
872
873
874
```

PatientID	PatientFirstName	PatientLastName	Gender	Address	PhoneNumber	dateofBirth	EmergencyContactName
PAT002	Jane	Smith	Female	456 Elm St	987-654-3210	1995-06-01	Bob Smith

Output

#	Time	Action	Message	Duration / Fetch
37	17:36:03	INSERT INTO PatientInformation (PatientID,PatientFirstName,PatientLastName,Gender,Address,PhoneNumber,DateOfBirth,Emer...	Error Code: 1062. Duplicate entry 'PAT002' for key 'patientinformation.PRIMARY'	0.031 sec
38	17:36:25	SELECT * FROM PatientInformation LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
39	17:40:15	SELECT * FROM PatientInformation WHERE PatientID IN ('PAT001', 'PAT002', 'PAT003') LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec

11. Aggregate function

A) Count function

How many appointments were made on each date?

The screenshot shows a SQL query in a text editor and its results in a grid. The query is:

```
865
866 • SELECT AppointmentDate, COUNT(*) AS TotalAppointments
867 FROM Appointments
868 GROUP BY AppointmentDate;
869
870
871
872
873
874
```

The results grid shows the following data:

AppointmentDate	TotalAppointments
2024-10-21	1
2024-10-22	1
2024-10-23	1
2024-10-24	1

The bottom panel shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
38	17:36:25	SELECT * FROM PatientInformation LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
39	17:40:15	SELECT * FROM PatientInformation WHERE PatientID IN (PAT001, PAT002, PAT003) LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
40	17:48:33	SELECT AppointmentDate, COUNT(*) AS TotalAppointments FROM Appointments GROUP BY AppointmentDate LIMIT 0, 1000	4 row(s) returned	0.000 sec / 0.000 sec

B) Average function

What is the average number of appointments per day?

The screenshot shows a SQL query in a text editor and its results in a grid. The query is:

```
885 • SELECT AVG(AppointmentCount) AS AverageAppointments
886 FROM (
887 SELECT AppointmentDate, COUNT(*) AS AppointmentCount
888 FROM Appointments
889 GROUP BY AppointmentDate
890 ) AS DailyAppointments;
891
```

The results grid shows the following data:

AverageAppointments
1.0000

The bottom panel shows the execution log with the following entries:

#	Time	Action	Message	Duration / Fetch
45	17:59:13	SELECT AVG(Amount) AS AverageBill FROM Bills LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
46	18:00:10	SELECT AVG(YEAR(CURDATE()) - YEAR(DateOfBirth)) AS AverageAge FROM PatientInformation LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
47	18:00:35	SELECT AVG(AppointmentCount) AS AverageAppointments FROM (SELECT AppointmentDate, COUNT(*) AS AppointmentCo...	1 row(s) returned	0.000 sec / 0.000 sec

C) Sum function

What is the total amount of bills for all patients?

891

892 •

```
SELECT SUM(Amount) AS TotalBillAmount
```

893

```
FROM Bills;
```

Result Grid

Filter Rows: Exports Wrap Cell Content:

TotalBillAmount

350.00

Result Grid

Result 33 ×

Output

Action Output

#	Time	Action	Message	Duration / Fetch
46	18:00:10	<pre>SELECT AVG(YEAR(CURDATE()) - YEAR(DateOfBirth)) AS AverageAge FROM PatientInformation LIMIT 0, 1000</pre>	1 row(s) returned	0.000 sec / 0.000 sec
47	18:00:35	<pre>SELECT AVG(AppointmentCount) AS AverageAppointments FROM (SELECT AppointmentDate, COUNT(*) AS AppointmentCo...</pre>	1 row(s) returned	0.000 sec / 0.000 sec
48	18:04:15	<pre>SELECT SUM(Amount) AS TotalBillAmount FROM Bills LIMIT 0, 1000</pre>	1 row(s) returned	0.000 sec / 0.000 sec

12.Group By clause

Count staff members by department.

892 • SELECT SUM(Amount) AS TotalBillAmount

893 FROM Bills;

894

895 • SELECT DepartmentID, COUNT(*) as StaffCount

896 FROM Staff

897 GROUP BY DepartmentID;

Result Grid

Filter Rows:

Export:

Wrap Call Contents:

DepartmentID	StaffCount
DEPT001	1
DEPT002	1
DEPT003	1

Result Grid

Result 34

Output

Action Output

#	Time	Action	Message	Duration / Fetch
47	18:00:35	SELECT AVG(AppointmentCount) AS AverageAppointments FROM (SELECT AppointmentDate, COUNT(*) AS AppointmentCo...	1 row(s) returned	0.000 sec / 0.000 sec
48	18:04:15	SELECT SUM(Amount) AS TotalBillAmount FROM Bills LIMIT 0, 1000	1 row(s) returned	0.000 sec / 0.000 sec
49	18:11:13	SELECT DepartmentID, COUNT(*) as StaffCount FROM Staff GROUP BY DepartmentID LIMIT 0, 1000	3 row(s) returned	0.000 sec / 0.000 sec

Read Only

Context Help

Snippets

What is the total quantity of items in the inventory for each department?

898

899 •

```
SELECT DepartmentID, SUM(QuantityInStock) AS TotalQuantity
```

900

```
FROM Inventory
```

901

```
GROUP BY DepartmentID;
```

Result Grid

Filter Rows

Export

Wrap Cell Content

DepartmentID	TotalQuantity
DEPT001	100
DEPT002	50
DEPT003	200

Result 35

Output

Action Output

#	Time	Action	Message
✓ 48	18:04:15	SELECT SUM(Amount) AS TotalBillAmount FROM Bills LIMIT 0, 1000	1 row(s) returned
✓ 49	18:11:13	SELECT DepartmentID, COUNT(*) as StaffCount FROM Staff GROUP BY DepartmentID LIMIT 0, 1000	3 row(s) returned
✓ 50	01:09:53	SELECT DepartmentID, SUM(QuantityInStock) AS TotalQuantity FROM Inventory GROUP BY DepartmentID LIMIT 0, 1000	3 row(s) returned

What are the highest and lowest bill amounts for each patient?

```
903 • SELECT PatientID, MAX(Amount) AS MaxBill, MIN(Amount) AS MinBill
904 FROM Bills
905 GROUP BY PatientID;
```

Result Grid

PatientID	MaxBill	MinBill
PAT001	100.00	100.00
PAT002	200.00	200.00
PAT003	50.00	50.00

Result 36 x

Output

Action Output

#	Time	Action	Message
50	01:09:53	SELECT DepartmentID, SUM(QuantityInStock) AS TotalQuantity FROM Inventory GROUP BY DepartmentID LIMIT 0, 1000	3 row(s) returned
51	01:16:09	SELECT InsuranceProvider, MAX(ClaimAmount) AS MaxClaim FROM InsuranceClaims GROUP BY InsuranceProvider LIMIT 0, ...	Error Code: 1054. Unknown column 'Cl
52	01:17:23	SELECT PatientID, MAX(Amount) AS MaxBill, MIN(Amount) AS MinBill FROM Bills GROUP BY PatientID LIMIT 0, 1000	3 row(s) returned

13. Like operator

Which patients have first names starting with 'J'?

```
907 • SELECT *
908 FROM PatientInformation
909 WHERE PatientFirstName LIKE 'J%';
```

Result Grid

PatientID	PatientFirstName	PatientLastName	Gender	Address	PhoneNumber	dateofbirth	EmergencyContactName
PAT002	Jane	Smith	Female	456 Elm St	987-654-3210	1995-06-01	Bob Smith

PatientInformation 37 x

Output

Action Output

#	Time	Action	Message
51	01:16:09	SELECT InsuranceProvider, MAX(ClaimAmount) AS MaxClaim FROM InsuranceClaims GROUP BY InsuranceProvider LIMIT 0, ...	Error Code: 1054. Unknown column 'Cl
52	01:17:23	SELECT PatientID, MAX(Amount) AS MaxBill, MIN(Amount) AS MinBill FROM Bills GROUP BY PatientID LIMIT 0, 1000	3 row(s) returned
53	01:21:33	SELECT * FROM PatientInformation WHERE PatientFirstName LIKE 'J%' LIMIT 0, 1000	1 row(s) returned

Which staff members have last names ending with 'son'?

```

911 • SELECT *
912 FROM Staff
913 WHERE LastName LIKE '%son';

```

StaffID	FirstName	LastName	Position	DepartmentID
STF003	Bob	Johnson	Admin	DEPT003

Staff 38 x

Output

Action Output

#	Time	Action	Message
52	01:17:23	SELECT PatientID, MAX(Amount) AS MaxBill, MIN(Amount) AS MinBill FROM Bills GROUP BY PatientID LIMIT 0, 1000	3 row(s) returned
53	01:21:33	SELECT * FROM PatientInformation WHERE PatientFirstName LIKE 'J%' LIMIT 0, 1000	1 row(s) returned
54	01:25:37	SELECT * FROM Staff WHERE LastName LIKE '%son' LIMIT 0, 1000	1 row(s) returned

Which patients have 'Smith' in their first or last name?

```

915 • SELECT *
916 FROM PatientInformation
917 WHERE PatientFirstName LIKE '%Smith%' OR PatientLastName LIKE '%Smith%';

```

PatientID	PatientFirstName	PatientLastName	Gender	Address	PhoneNumber	dateofbirth	EmergencyContactName
PAT002	Jane	Smith	Female	456 Elm St	987-654-3210	1995-06-01	Bob Smith

PatientInformation 39 x

Output

Action Output

#	Time	Action	Message
53	01:21:33	SELECT * FROM PatientInformation WHERE PatientFirstName LIKE 'J%' LIMIT 0, 1000	1 row(s) returned
54	01:25:37	SELECT * FROM Staff WHERE LastName LIKE '%son' LIMIT 0, 1000	1 row(s) returned
55	01:27:10	SELECT * FROM PatientInformation WHERE PatientFirstName LIKE '%Smith%' OR PatientLastName LIKE '%Smith%' LIMIT 0, 1...	1 row(s) returned

Which inventory items have names starting with 'Med'?

```

919 • SELECT *
920 FROM Inventory
921 WHERE ItemName LIKE 'Med%';

```

Result Grid

ItemID	ItemName	QuantityInStock	DepartmentID
ITM003	Medication	200	DEPT003

Inventory 40 x

Output

Action Output

#	Time	Action	Message
✓ 54	01:25:37	SELECT * FROM Staff WHERE LastName LIKE '%son' LIMIT 0, 1000	1 row(s) returned
✓ 55	01:27:10	SELECT * FROM PatientInformation WHERE PatientFirstName LIKE '%Smith%' OR PatientLastName LIKE '%Smith%' LIMIT 0, 1...	1 row(s) returned
✓ 56	01:29:14	SELECT * FROM Inventory WHERE ItemName LIKE 'Med%' LIMIT 0, 1000	1 row(s) returned

Which insurance claims have 'CL' somewhere in their IDs?

```

923 • SELECT *
924 FROM InsuranceClaims
925 WHERE ClaimID LIKE '%CL%';

```

Result Grid

ClaimID	BillID	InsuranceProvider	ClaimDate	Status
CLM001	BILL001	HealthCare Inc.	2024-10-21	Pending
CLM002	BILL002	Medicare	2024-10-22	Approved
CLM003	BILL003	UnitedHealth	2024-10-23	Denied

InsuranceClaims 41 x

Output

Action Output

#	Time	Action	Message
✓ 55	01:27:10	SELECT * FROM PatientInformation WHERE PatientFirstName LIKE '%Smith%' OR PatientLastName LIKE '%Smith%' LIMIT 0, 1...	1 row(s) returned
✓ 56	01:29:14	SELECT * FROM Inventory WHERE ItemName LIKE 'Med%' LIMIT 0, 1000	1 row(s) returned
✓ 57	01:31:24	SELECT * FROM InsuranceClaims WHERE ClaimID LIKE '%CL%' LIMIT 0, 1000	3 row(s) returned

14.Union

Can you list the names of both patients and staff members?


```

929 • SELECT PatientFirstName AS Name, PatientLastName AS Surname
930 FROM PatientInformation
931 UNION
932 SELECT FirstName AS Name, LastName AS Surname
933 FROM Staff;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: [IA](#)

	Name	Surname
▶	Jane	Smith
	John	Doe
	Bob	Johnson

Result 45 x

Output

Action Output

#	Time	Action	Message
59	01:33:58	SELECT * FROM Appointments WHERE AppointmentDate LIKE '2024-21%' LIMIT 0, 1000	0 row(s) returned
60	01:34:05	SELECT * FROM Appointments WHERE AppointmentDate LIKE '2024-10%' LIMIT 0, 1000	4 row(s) returned
61	01:38:28	SELECT PatientFirstName AS Name, PatientLastName AS Surname FROM PatientInformation UNION SELECT FirstName AS N...	3 row(s) returned

Which rooms are mentioned either in the appointments or in the rooms table?

```

934 • SELECT RoomNumber FROM Appointments
935 UNION
936 SELECT RoomNumber FROM Rooms;
937
938

```

Result Grid | Filter Rows: | Export: | Wrap Cell Contents: [IA](#)

	RoomNumber
▶	RM101
	RM202
	RM303
	RM404
	R101
	R102
	R103

Result 46 x

Output

Action Output

#	Time	Action	Message
60	01:34:05	SELECT * FROM Appointments WHERE AppointmentDate LIKE '2024-10%' LIMIT 0, 1000	4 row(s) returned
61	01:38:28	SELECT PatientFirstName AS Name, PatientLastName AS Surname FROM PatientInformation UNION SELECT FirstName AS N...	3 row(s) returned
62	01:42:55	SELECT RoomNumber FROM Appointments UNION SELECT RoomNumber FROM Rooms	7 row(s) returned

Can you show all appointments for October and November 2024?

```

937 • SELECT AppointmentID, AppointmentDate FROM Appointments WHERE AppointmentDate LIKE '2024-10%'
938 UNION
939 SELECT AppointmentID, AppointmentDate FROM Appointments WHERE AppointmentDate LIKE '2024-11%';

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	AppointmentID	AppointmentDate
▶	APT001	2024-10-21
	APT002	2024-10-22
	APT003	2024-10-23
	APT004	2024-10-24

Result 47 x

Output

Action Output

#	Time	Action	Message
✓ 62	01:42:55	SELECT RoomNumber FROM Appointments UNION SELECT RoomNumber FROM Rooms	7 row(s) returned
✗ 63	01:47:38	SELECT StaffID, FirstName, LastName FROM Staff WHERE Status = 'Active' UNION SELECT StaffID, FirstName, LastName FR...	Error Code: 1054. Unknown c
✓ 64	01:48:22	SELECT AppointmentID, AppointmentDate FROM Appointments WHERE AppointmentDate LIKE '2024-10%' UNION SELECT Ap...	4 row(s) returned

15. Joins

Which patients have or do not have appointments?

```

941 • SELECT Patients.PatientID, Patients.PatientFirstName, Appointments.AppointmentDate
942 FROM PatientInformation AS Patients
943 LEFT JOIN Appointments
944 ON Patients.PatientID = Appointments.PatientID;

```

Result Grid | Filter Rows: | Export: | Wrap Cell Content: [IA](#)

	PatientID	PatientFirstName	AppointmentDate
▶	PAT002	Jane	2024-10-22

Result 48 x

Output

Action Output

#	Time	Action	Message
✗ 63	01:47:38	SELECT StaffID, FirstName, LastName FROM Staff WHERE Status = 'Active' UNION SELECT StaffID, FirstName, LastName FR...	Error Code: 1054. Unknown
✓ 64	01:48:22	SELECT AppointmentID, AppointmentDate FROM Appointments WHERE AppointmentDate LIKE '2024-10%' UNION SELECT Ap...	4 row(s) returned
✓ 65	02:00:32	SELECT Patients.PatientID, Patients.PatientFirstName, Appointments.AppointmentDate FROM PatientInformation AS Patients LE...	1 row(s) returned

How would you modify the query to include only appointments for patients with a specific last name?

```
945
946 • SELECT
947     Appointments.AppointmentID,
948     Appointments.AppointmentDate,
949     PatientInformation.PatientFirstName,
950     PatientInformation.PatientLastName
951 FROM
952     PatientInformation
953 RIGHT JOIN Appointments
954 ON PatientInformation.PatientID = Appointments.PatientID;
955
956
```

Result Grid

	AppointmentID	AppointmentDate	PatientFirstName	PatientLastName
▶	APT001	2024-10-21	NULL	NULL
	APT002	2024-10-22	Jane	Smith
	APT003	2024-10-23	NULL	NULL
	APT004	2024-10-24	NULL	NULL

Result 62 x

Output

Action Output

#	Time	Action	Message
86	02:31:52	SELECT * FROM Rooms WHERE RoomNumber NOT IN (SELECT RoomNumber FROM Appointments) LIMIT 0, 1000	3 row(s) returned
87	02:32:04	SELECT Rooms.RoomNumber, Appointments.AppointmentID, Appointments.AppointmentDate, PatientInformation.Patient...	1 row(s) returned
88	02:35:37	SELECT Appointments.AppointmentID, Appointments.AppointmentDate, PatientInformation.PatientFirstName, PatientInfor...	4 row(s) returned

16.Sub query

Find rooms with no appointments

```

962 );
963 SELECT *
964 FROM Rooms
965 WHERE RoomNumber NOT IN (
966     SELECT RoomNumber
967     FROM Appointments
968 );

```

Result Grid

RoomNumber	DepartmentID	RoomType
R101	DEPT001	Examination
R102	DEPT002	Operating
R103	DEPT003	Recovery
NULL	NULL	NULL

Rooms 64 x

Output

Action Output

#	Time	Action	Message
88	02:35:37	SELECT Appointments.AppointmentID, Appointments.AppointmentDate, PatientInformation.PatientFirstName, PatientInfor...	4 row(s) returned
89	02:43:05	SELECT * FROM PatientInformation WHERE PatientID IN (SELECT PatientID FROM Appointments WHERE AppointmentD...	0 row(s) returned
90	02:43:53	SELECT * FROM Rooms WHERE RoomNumber NOT IN (SELECT RoomNumber FROM Appointments) LIMIT 0, 1000	3 row(s) returned

Retrieve staff members with a specific position

```

989 SELECT *
990 FROM Staff
991 WHERE Position = (
992     SELECT Position
993     FROM Staff
994     WHERE StaffID = 'STF001'
995 );

```

Result Grid

StaffID	FirstName	LastName	Position	DepartmentID
STF001	John	Doe	Doctor	DEPT001
NULL	NULL	NULL	NULL	NULL

Staff 68 x

Output

Action Output

#	Time	Action	Message
97	02:56:07	USE RR	0 row(s) affected
98	02:56:24	SELECT * FROM (SELECT a.PatientID, p.PatientFirstName, a.AppointmentDate FROM Appointments a JOIN PatientInformatio...	0 row(s) returned
99	02:59:31	SELECT * FROM Staff WHERE Position = (SELECT Position FROM Staff WHERE StaffID = 'STF001') LIMIT 0, 1000	1 row(s) returned

Retrieve departments with staff members

```

1003
1004 • SELECT *
1005 FROM Departments
1006 WHERE DepartmentID IN (
1007     SELECT DepartmentID
1008     FROM Staff
1009 );

```

Result Grid

DepartmentID	DepartmentName
DEPT003	Administration
DEPT001	Cardiology
DEPT002	Neurology
NULL	NULL

Departments 73 x

Output

Action Output

#	Time	Action	Message
✓ 102	03:02:07	SELECT * FROM Staff WHERE Position = (SELECT Position FROM Staff WHERE StaffID = 'STF001') LIMIT 0, 1000	1 row(s) returned
✓ 103	03:02:11	SELECT * FROM Staff WHERE DepartmentID = (SELECT DepartmentID FROM Departments WHERE DepartmentName = '...	1 row(s) returned
✓ 104	03:02:46	SELECT * FROM Departments WHERE DepartmentID IN (SELECT DepartmentID FROM Staff) LIMIT 0, 1000	3 row(s) returned

17.Views

```

1024 • CREATE VIEW PatientAppointments AS
1025 SELECT p.PatientFirstName, p.PatientLastName, a.AppointmentDate
1026 FROM PatientInformation p
1027 JOIN Appointments a ON p.PatientID = a.PatientID;
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037

```


Output

Action Output

#	Time	Action	Message
✓ 105	03:05:37	SELECT * FROM Inventory WHERE QuantityInStock < (SELECT AVG(QuantityInStock) FROM Inventory) LIMIT 0, 1000	2 row(s) returned
✓ 106	03:08:15	SELECT * FROM Departments WHERE DepartmentID = (SELECT DepartmentID FROM Rooms WHERE RoomType = 'Exa...	1 row(s) returned
✓ 107	03:13:31	CREATE VIEW PatientAppointments AS SELECT p.PatientFirstName, p.PatientLastName, a.AppointmentDate FROM PatientInfo...	0 row(s) affected

1028


1029 • `SELECT * FROM PatientAppointments;`

Result Grid   Filter Rows: Exports:  Wrap Cell Content: 

	PatientFirstName	PatientLastName	AppointmentDate
▶	Jane	Smith	2024-10-22

PatientAppointments 76 x

Output

 Action Output ▼

	#	Time	Action	Message
✓	106	03:08:15	SELECT * FROM Departments WHERE DepartmentID = (SELECT DepartmentID FROM Rooms WHERE RoomType = 'Exa...	1 row(s) returned
✓	107	03:13:31	CREATE VIEW PatientAppointments AS SELECT p.PatientFirstName, p.PatientLastName, a.AppointmentDate FROM PatientInfo...	0 row(s) affected
✓	108	03:14:31	SELECT * FROM PatientAppointments LIMIT 0, 1000	1 row(s) returned

