

Cab Booking System with SQL

This presentation outlines a robust, data-driven solution for a growing urban sector. We will explore the importance of automation and how SQL powers a seamless, efficient cab booking experience, from real-time assignments to insightful analytics.



Problem Statement:

Addressing Inefficiencies in Traditional Cab Booking Systems



Lack of Centralized Data Access

Customer, driver, cab, and booking records are often stored separately, making it difficult to track rides, manage operations, and maintain historical data.



Manual Booking Systems Are Inefficient

Bookings handled through phone calls or manual entries lead to delays, data inconsistency, duplication, and higher chances of human error.



No Real-Time Analytics or Reporting

Organizations lack real-time insights into booking trends, active trips, idle cabs, cancellations, and driver performance, resulting in poor decision-making.

Objective:

Track Trip Lifecycle from Booking to Completion

- Monitor the complete workflow: customer booking → driver assignment → trip execution → fare generation → feedback collection.
- Maintain accurate time-stamped logs for bookings, trip start/end times, distances, fares, and customer interactions.

Perform Behavioural & Revenue Analytics

- Identify patterns such as frequent cancellations, top-performing drivers, and high-demand booking periods.
- Analyze revenue trends through trip fares, customer spending, and driver earnings.

Enable Feedback and Quality Monitoring

- Capture customer ratings and feedback for every trip to evaluate service satisfaction.
- Analyze driver performance using average ratings and identify low-rated drivers for corrective action.

Support Operational Decision-Making

- Assist management in optimizing cab availability, driver allocation, and trip distribution based on real data.
- Enable data-driven reward/penalty decisions for drivers and customers based on performance indicators.

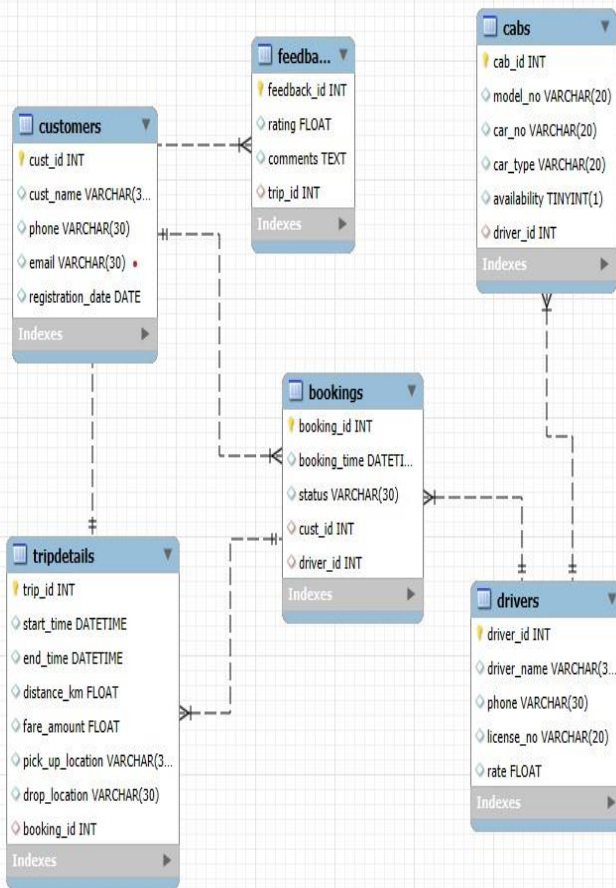
Entity-Relationship (ER) Diagram

A visual layout of our database design.

- One-to-many relationships: One customer can make many bookings, and one driver can complete many rides.

- Key tables include: Customers, Drivers, Cabs, and Bookings.

- Primary and foreign keys are explicitly defined for clarity



Retrieve Cab Details That Are Currently Available

Objective:

Identify all cabs that are currently marked as available for rides.

```
SELECT *  
FROM cabs  
WHERE availability =  
'True';
```

Result Grid		Filter Rows:		Edit:		Export/Import	
	cab_id	model_no	car_no	car_type	availability	driver_id	
	203	Honda City	MH03EF3456	Sedan	0	103	
	206	Mahindra XUV	MH06KL6789	SUV	0	106	
	210	Honda Amaze	MH10ST0123	Sedan	0	110	
	215	Hyundai Creta	MH15CD5678	SUV	0	115	
▶	218	Tata Altroz	MH18IJ8901	Hatchback	0	118	
•	NULL	NULL	NULL	NULL	NULL	NULL	

Insight:

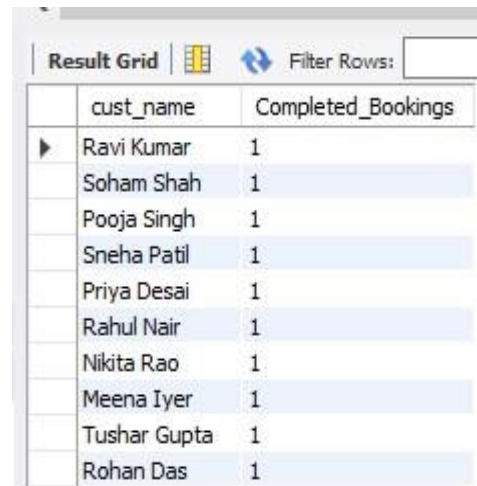
Shows the active fleet available for immediate allocation — useful for operational readiness.

Customers With Most Completed Bookings.

Objective:

Identify loyal or highly active customers.

```
SELECT cust_name, COUNT(*) AS  
Completed_Bookings  
FROM customers, bookings  
WHERE customers.cust_id =  
bookings.cust_id  
AND bookings.status = 'Completed'  
GROUP BY customers.cust_name  
ORDER BY Completed_Bookings;
```



The screenshot shows a 'Result Grid' window with a 'Filter Rows' input field. The grid contains two columns: 'cust_name' and 'Completed_Bookings'. There are 11 rows of data, each with a customer name and the value '1'. The rows are: Ravi Kumar, Soham Shah, Pooja Singh, Sneha Patil, Priya Desai, Rahul Nair, Nikita Rao, Meena Iyer, Tushar Gupta, and Rohan Das.

cust_name	Completed_Bookings
Ravi Kumar	1
Soham Shah	1
Pooja Singh	1
Sneha Patil	1
Priya Desai	1
Rahul Nair	1
Nikita Rao	1
Meena Iyer	1
Tushar Gupta	1
Rohan Das	1

Insight:

Shows top engaged users—ideal for loyalty programs or personalized discounts.

Customer Who Completed the Maximum Number of Bookings

Objective:

To identify the customer with the highest booking completion count.
This helps in determining the most loyal or frequent user.

```
SELECT driver_id, COUNT(*) AS  
Completed_Bookings  
FROM bookings  
WHERE status = 'Completed'  
GROUP BY driver_id  
ORDER BY Completed_Bookings  
DESC;
```

Result Grid		Filter Rows:
	driver_id	Completed_Bookings
▶	101	1
	103	1
	104	1
	106	1
	108	1
	109	1
	110	1
	112	1
	113	1
	115	1

Insight:

Shows the top-performing customer based on completed rides—important for loyalty program offerings.

Driver Who Completed the Maximum Bookings

Objective:

To find the driver with the highest number of completed bookings.
Useful for driver performance evaluation and incentive planning.

```
SELECT driver_name,  
COUNT(*) AS  
Completed_Bookings  
FROM bookings, drivers  
WHERE bookings.driver_id =  
drivers.driver_id  
AND status = 'Completed'  
GROUP BY driver_name  
ORDER BY  
Completed_Bookings DESC;
```

	Driver_name	Completed_Bookings
▶	Amit Sharma	1
	Farhan Ali	1
	Jatin Desai	1
	Karan Tiwari	1
	Mahesh Iyer	1
	Manoj Singh	1
	Nikhil Sen	1
	Nilesh Joshi	1
	Prakash Naik	1
	Raj Singh	1

Insight:

Identifies star drivers who complete the most trips, contributing greatly to business revenue.

Top 6 Customers by Completed Bookings

Objective:

To identify the three most engaged customers based on completed rides.
Useful for retention and targeted promotional offers.

```
select cust_name,status,count(*)  
as Completed_Bookings  
from bookings,customers  
where bookings.cust_id =  
customers.cust_id and  
status = 'Completed'  
group by cust_name  
limit 6
```

	cust_name	status	Completed_Bookings
▶	Ravi Kumar	Completed	1
	Soham Shah	Completed	1
	Pooja Singh	Completed	1
	Sneha Patil	Completed	1
	Priya Desai	Completed	1
	Rahul Nair	Completed	1

Insight:

Highlights premium customers who frequently use the service.

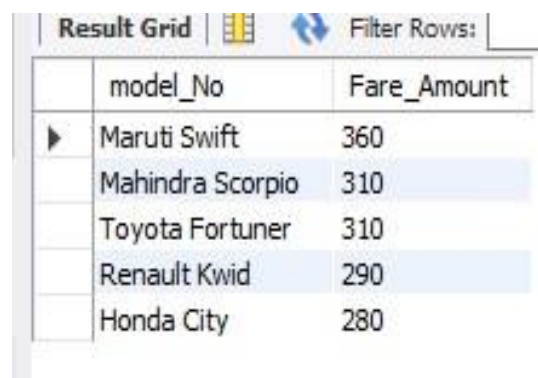
Cab Model with Highest Total Fare Generated

Objective:

To find which cab model contributes most to total revenue.

This helps in evaluating vehicle performance and future purchasing decisions.

```
SELECT model_No,  
MAX(tripdetails.fare_amount) AS  
Fare_Amount  
FROM cabs, tripdetails, bookings  
WHERE bookings.booking_id =  
tripdetails.booking_id  
AND bookings.driver_id =  
cabs.driver_id  
GROUP BY cabs.model_no  
ORDER BY Fare_Amount DESC  
LIMIT 6;
```



The screenshot shows a 'Result Grid' window with a table containing 6 rows of data. The columns are 'model_No' and 'Fare_Amount'. The rows are: Maruti Swift (360), Mahindra Scorpio (310), Toyota Fortuner (310), Renault Kwid (290), and Honda City (280). The first row is highlighted with a blue background.

	model_No	Fare_Amount
▶	Maruti Swift	360
	Mahindra Scorpio	310
	Toyota Fortuner	310
	Renault Kwid	290
	Honda City	280

Insight:

Shows the best-performing cab model in terms of revenue generation.

Customer Who Spent the Most Money

Objective:

To find the highest-paying customer based on completed ride fares.
Useful for strategic marketing and loyalty rewards.

```
SELECT cust_name, status,  
SUM(tripdetails.fare_amount) AS  
Total_SpentAmt  
  
FROM bookings, customers,  
tripdetails  
  
WHERE bookings.cust_id =  
customers.cust_id  
  
AND bookings.booking_id =  
tripdetails.booking_id  
  
AND bookings.status = 'Completed'  
  
GROUP BY cust_name  
ORDER BY Total_SpentAmt DESC;
```

	cust_name	status	Total_SpentAmt
▶	Pooja Singh	Completed	360
	Rahul Nair	Completed	310
	Soham Shah	Completed	280
	Aishwarya Sen	Completed	275
	Neha Kulkarni	Completed	260
	Priya Desai	Completed	250
	Ravi Kumar	Completed	240
	Tushar Gupta	Completed	240
	Divya Pillai	Completed	220
	Sneha Patil	Completed	210

Insight:

Shows the biggest contributor to revenue among all customers.

Revenue Generated by Each Driver

Objective:

To calculate earnings per driver from completed trips.
Supports performance analysis and incentive systems.

```
SELECT driver_name,  
SUM(tripdetails.fare_amount) AS  
Total_Revenue  
FROM bookings, drivers, tripdetails  
WHERE bookings.driver_id =  
drivers.driver_id  
AND bookings.booking_id =  
tripdetails.booking_id  
GROUP BY driver_name  
ORDER BY Total_Revenue DESC  
LIMIT 5;
```

	driver_name	Total_Revenue
▶	Vikram Yadav	360
	Arjun Kapoor	310
	Sunil Shetty	310
	Ravi Chauhan	290
	Amit Sharma	280

Insight:

Identifies the highest revenue-generating driver.

Conclusion:

This Cab Booking Analysis project demonstrated the complete end-to-end workflow of building a real-world analytical database system — from designing structured tables to generating meaningful SQL insights. The project successfully transformed raw operational data into actionable intelligence for decision-making.

Meaningful Business Insights

Extracted key insights from customer behaviour, driver performance, cab utilization, trip trends, and revenue distribution using SQL queries.

Real-Time Booking Logic Representation

Simulated real-world cab booking operations using well-structured tables for customers, drivers, cabs, bookings, trip details, and feedback.

This logical model supports automated driver assignment, trip tracking, fare calculation, and customer experience assessment.

Creation of Analytical SQL Queries

Developed a wide range of SQL queries — from basic retrieval to advanced nested subqueries — that mirror actual business challenges such as identifying top customers, most revenue-generating drivers, cancellation patterns, and peak demand trips.

Strong Foundation for Further Expansion

The project establishes a scalable database structure that can be extended with features such as dynamic pricing, ride-sharing logic, demand prediction, or real-time dashboards.

Thank You!

We appreciate your time and attention and hope this presentation provided valuable insights into our automated cab booking system.

