

# **LA ATELIER: A place to ride your DREAM.**

## **Subject name:**

Database Design and SQL (CSD 2206)

## **Project Name:**

La Atelier

## **Term:**

Winter 2024

## **Professor Name:**

Mehernoush Ashrafi

## **Team Members:**

- Arda Yesilirmak (C0923184)
- Process Parajuli (C0906709)
- Rajesh Haluwai (C0924930)
- Raju Tamang (C0915472)
- Saugat Bikram Shah (C0927647)

## **Table of Contents**

1. Company Overview
2. Company rules and regulations
3. Products
4. Product Attributes
5. Company Invoice
6. Many to Many Relationships
7. Many to Many resolved Relationships
8. ER language.
9. Relational Data
10. Queries

# Company Overview:

La Atelier car rental service is customer-centric, operating with a robust management system overseeing every aspect of the rental process. Renters lease directly from the owners, making it easy to provide essential details for seamless communication and safety. Meanwhile owners trust us to manage their vehicles efficiently, ensuring transparency and maximizing returns. Reservations, identified by unique IDs, offer accurate and reliable booking experiences for both the renter and the owner. Our diverse fleet meets industry standards for quality and safety. Our payment system tracks transactions, contracts ensure clarity, and dedicated employees prioritize customer service. Our comprehensive insurance system provides peace of mind for all parties. Our system optimizes the rental experience, prioritizing satisfaction, transparency, and efficiency.

# Rules and Regulations:

- Renters need to provide their full name, contact details, and a valid driver's license to start the rental procedure.
- Renters are required to agree to the terms of the reservation, including rental duration, rate, and any applicable charges.
- Renters must use a valid credit card as payment method to confirm their reservation.
- Vehicle owners must keep their cars in good condition and have the appropriate insurance coverage.
- Owners must accurately disclose vehicle details such as make, model, license, plate number, manufacturing date, transmission type, engine, and tires.
- Each employee is assigned at least one car, and each car is overseen by 1 employee.
- Employees have the responsibility of managing reservations, allocating vehicles, and handling payment processing.
- Employees must verify a valid reservation and driver's license before releasing the vehicle.
- Employees need to inspect the vehicle's condition before and after the rental period.
- Timely and accurate processing and recording of payments are essential.
- A contract needs to be created for every reservation, detailing information such as reservation ID, rental period, price, and any extra fees.
- Vehicles must be insured, detailing the type of coverage and insurance provider.
- Renters must return the vehicle in its original condition, excluding regular wear and tear, and are liable for any damages incurred during the rental period, with repair expenses billed to their chosen payment method.
- Renters must return the vehicle to the designated location at the end of the rental period.
- Failure to comply with these regulations may result in reservation cancellation and additional fees.  
**(\*These guidelines are subject to updates and should be followed by all involved in the car rental process for a smooth and satisfactory experience.)**

## **Products:**

- CAR
  - 1. Honda
  - 2. BMW
  - 3. Merecedes
  - 4. Porche
  - 5. Toyota

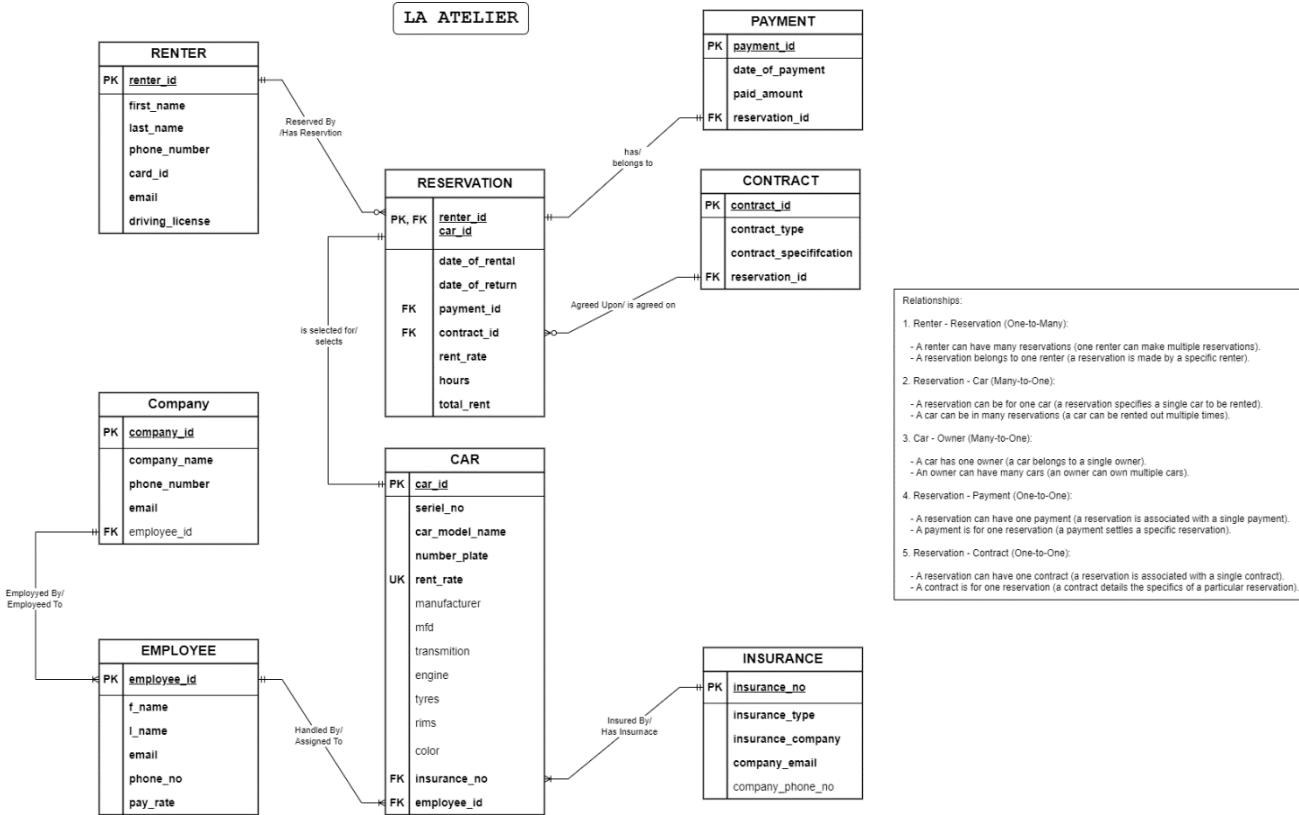
## **Product Attributes**

- Car id
- Serial Number
- Car model (name)
- Number plate
- Manufacturer
- Manufactured date (mfd)
- Transmission
- Engine
- Tyres
- Rims
- Rent rate (per hr)
- Color

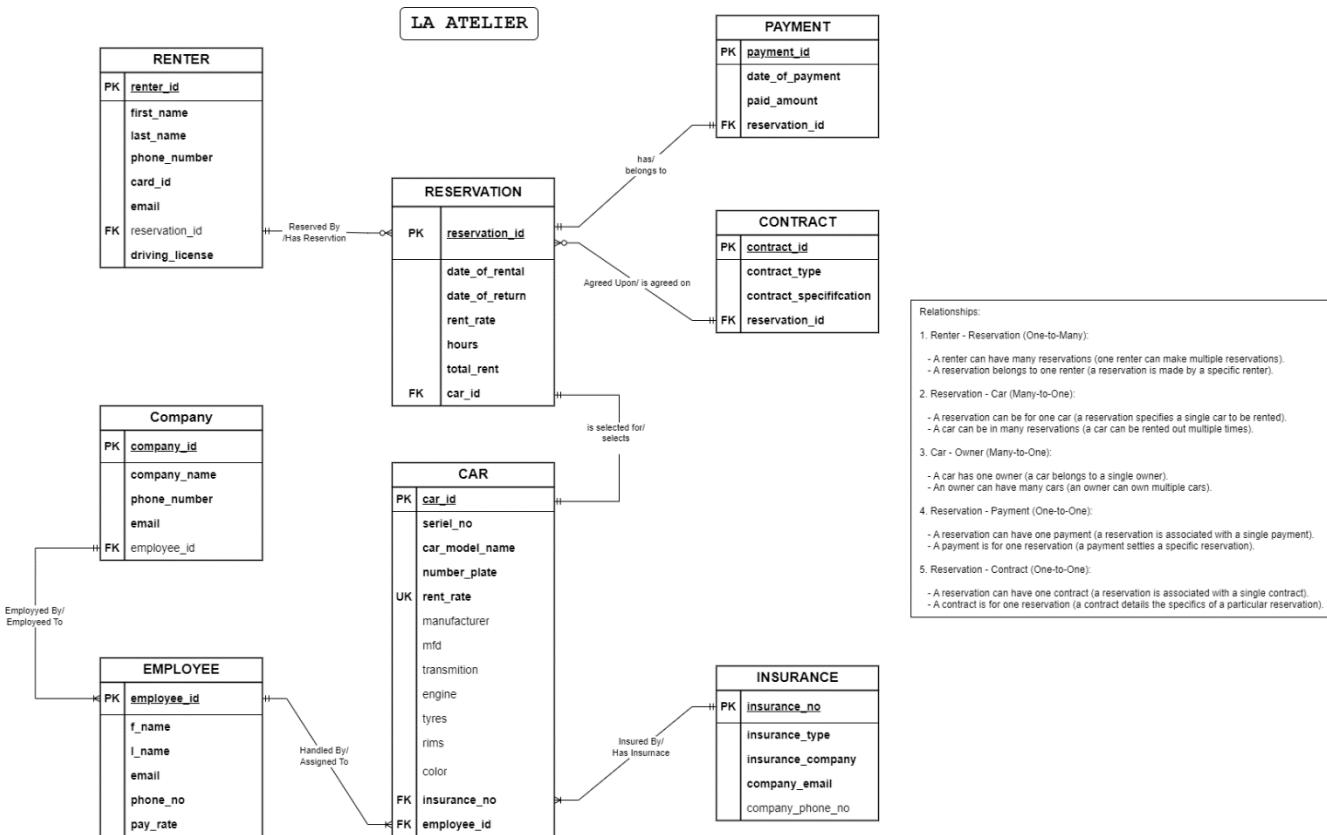
# Company Invoice:

LA ATELIER	
Bill Receipt	
Payment ID: 754718	Payment Date: 2024-04-10
Renter ID	756082
Car ID	451036
Model	BMW M760i xDrive
Reservation ID	504562
Contract ID	752086
Rental Start Date	2024-04-10
Rental End Date	2024-04-13
Rent (/ hr)	\$199.99 tax included)
Total Rent	\$14399.28 tax included)
Paid	\$14399.28 tax included)

# Many to Many Relationships(unresolved):



# Many-Many Relationships (resolved):



# **ER Language**

## 1. Renter - Reservation (One-to-Many):

- A renter can have many reservations (one renter can make 0 to multiple reservations).
- A reservation belongs to one renter (a reservation is made by a specific renter).

## 2. Reservation - Car (Many-to-One):

- A reservation can be for one car (a reservation specifies a single car to be rented).
- A car can be in many reservations (a car can be rented out multiple times).

## 3. Car - Owner (Many-to-One):

- A car has one owner (a car belongs to a single owner).
- An owner can have many cars (an owner can own multiple cars).

## 4. Reservation - Payment (One-to-One):

- A reservation can have one payment (a reservation is associated with a single payment).
- A payment is for one reservation (a payment settles a specific reservation).

## 5. Reservation - Contract (One-to-One):

- A reservation can have one contract (a reservation is associated with a single contract).
- A contract is for one reservation (a contract details the specifics of a particular reservation only)

# Relational Data Model

Car				
Key type	Optionality	Column name	Data Type	Length
PK	*	Car_id	int	10
	*	serial_no	varchar	50
	*	car_model_name	varchar	50
	*	number_plate	varchar	20
	O	manufacturer	varchar	20
	O	mfd	date	
	O	transmission	varchar	50
	O	engine	varchar	50
	O	tyres	varchar	50
	O	rims	varchar	50
UK	*	rent_rate	Int/decimal	(6,2)
	O	color	varchar	50
FK	*	Insurance_no	int	20
FK	*	employee_id	int	10

reservation				
Key type	Optionality	Column name	Data Type	Length
PK	*	reservation_id	int	10
	*	date_of_rental	date	
	*	date_of_return	date	
FK	*	rent_rate	Int/decimal	(6,2)
	*	hours	int	1000
	*	total_rent	int/decimal	(6,2)
FK	*	car_id	int	10

renter				
Key type	Optionality	Column name	Data Type	Length
PK	*	renter_id	int	10
	*	first_name	varchar	50
	*	last_name	varchar	50
	*	phone_number	varchar	20
	*	car_id	int	10
UK	*	email	varchar	100
FK	O	reservation_id	int	50
UK	*	driving_license	varchar	20

employee				
Key type	Optionality	Column name	Data Type	Length
PK	*	employee_id	int	10
	*	first_name	varchar	50
	*	last_name	varchar	50
UK	*	phone_number	varchar	20
UK	*	email	varchar	100
	*	pay_rate	decimal	(10,2)

company				
Key type	Optionality	Column name	Data Type	Length
PK	*	company_id	int	10
	*	company_name	varchar	50
	*	phone_number	varchar	20
UK	*	email	varchar	100
PK	O	employee_id	int	10

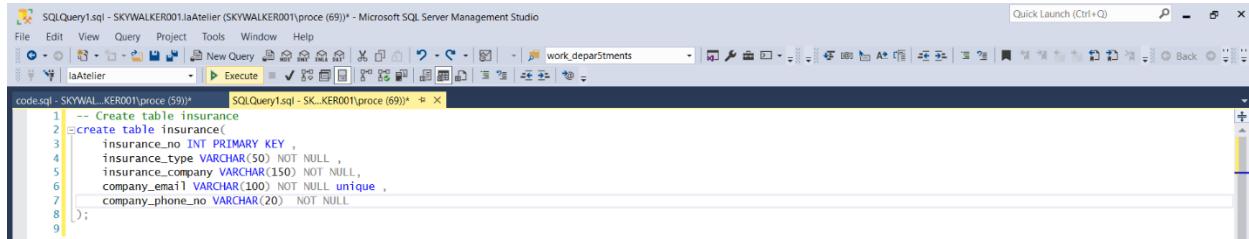
payment				
Key type	Optionality	Column name	Data Type	Length
PK	*	payment_id	int	10
	*	date_of_payment	date	
	*	paid_amount	Int / decimal	(6,2)
FK	*	reservation_id	int	10

contract				
Key type	Optionality	Column name	Data Type	Length
PK	*	contract_id	int	10
	*	contract_type	varchar	50
	*	contract_specifiaction	varchar	150
FK	*	reservation_id	int	10

insurance				
Key type	Optionality	Column name	Data Type	Length
PK	*	insurance_no	int	10
	*	insurance_type	varchar	50
	*	insurance_company	varchar	150
	*	company_email	varchar	100
	O	company_phone_no	Varchar	20

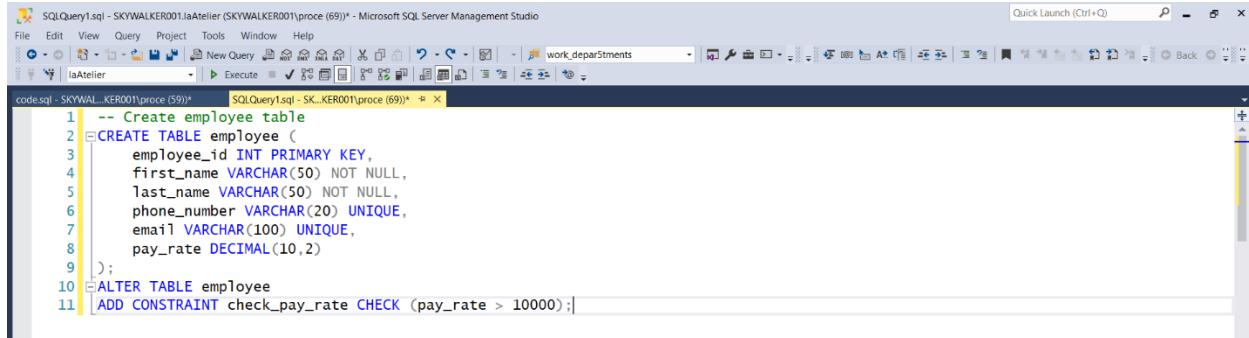
## Create database:

### 1) Create insurance table:



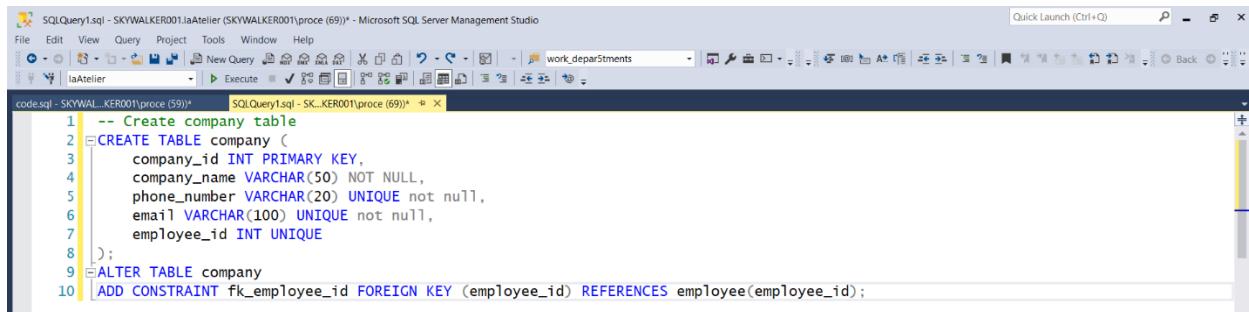
```
1 -- Create table insurance
2 CREATE TABLE insurance(
3     insurance_no INT PRIMARY KEY ,
4     insurance_type VARCHAR(50) NOT NULL ,
5     insurance_company VARCHAR(150) NOT NULL,
6     company_email VARCHAR(100) NOT NULL unique ,
7     company_phone_no VARCHAR(20) NOT NULL
8 );
```

### 2) Create employee table



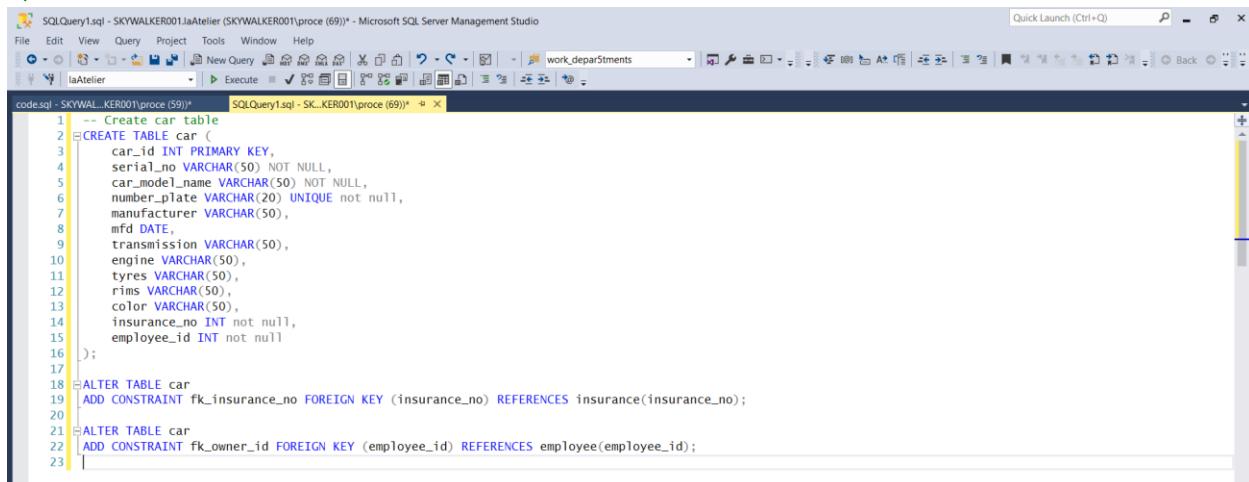
```
1 -- Create employee table
2 CREATE TABLE employee (
3     employee_id INT PRIMARY KEY ,
4     first_name VARCHAR(50) NOT NULL,
5     last_name VARCHAR(50) NOT NULL,
6     phone_number VARCHAR(20) UNIQUE ,
7     email VARCHAR(100) UNIQUE ,
8     pay_rate DECIMAL(10,2)
9 );
10 ALTER TABLE employee
11 ADD CONSTRAINT check_pay_rate CHECK (pay_rate > 10000);
```

### 3) Create company table



```
1 -- Create company table
2 CREATE TABLE company (
3     company_id INT PRIMARY KEY ,
4     company_name VARCHAR(50) NOT NULL,
5     phone_number VARCHAR(20) UNIQUE not null,
6     email VARCHAR(100) UNIQUE not null,
7     employee_id INT UNIQUE
8 );
9 ALTER TABLE company
10 ADD CONSTRAINT fk_employee_id FOREIGN KEY (employee_id) REFERENCES employee(employee_id);
```

### 4) Create car table



```
1 -- Create car table
2 CREATE TABLE car (
3     car_id INT PRIMARY KEY ,
4     serial_no VARCHAR(50) NOT NULL,
5     car_model_name VARCHAR(50) NOT NULL,
6     number_plate VARCHAR(20) UNIQUE not null,
7     manufacturer VARCHAR(50),
8     mfd DATE,
9     transmission VARCHAR(50),
10    engine VARCHAR(50),
11    tyres VARCHAR(50),
12    rims VARCHAR(50),
13    color VARCHAR(50),
14    insurance_no INT not null,
15    employee_id INT not null
16 );
17
18 ALTER TABLE car
19 ADD CONSTRAINT fk_insurance_no FOREIGN KEY (insurance_no) REFERENCES insurance(insurance_no);
20
21 ALTER TABLE car
22 ADD CONSTRAINT fk_owner_id FOREIGN KEY (employee_id) REFERENCES employee(employee_id);
```

## 5) Create the reservation table

```
1 -- Create the reservation table
2 CREATE TABLE reservation (
3     reservation_id INT PRIMARY KEY,
4     date_of_rental VARCHAR(50) NOT NULL,
5     date_of_return VARCHAR(50) NOT NULL,
6     rent_rate int NOT NULL,
7     hours INT not null,
8     total_rent int not null,
9     car_id int not null
10 );
```

## 6) Create renter table

```
1 -- Create renter table
2 CREATE TABLE renter (
3     renter_id INT PRIMARY KEY,
4     first_name VARCHAR(50) NOT NULL,
5     last_name VARCHAR(50) NOT NULL,
6     phone_number VARCHAR(20) UNIQUE NOT NULL,
7     card_id INT,
8     email VARCHAR(100) UNIQUE,
9     driving_license VARCHAR(20) UNIQUE,
10    reservation_id INT NOT NULL
11 );
12 ALTER TABLE renter
13 ADD constraint fk_reservation_id FOREIGN KEY (reservation_id) REFERENCES reservation(reservation_id);
```

## 7) Create payment table

```
1 -- Create payment table
2 CREATE TABLE payment (
3     payment_id INT PRIMARY KEY,
4     date_of_payment date,
5     paid int NOT NULL,
6     reservation_id INT
7 );
8 ALTER TABLE payment
9 ADD CONSTRAINT fk_payment_reservation
10 FOREIGN KEY (reservation_id) REFERENCES reservation(reservation_id);
```

## 8) create contract table

```
1 -- Create table contract
2 CREATE TABLE contract (
3     contract_id INT PRIMARY KEY,
4     contract_type VARCHAR(50) NOT NULL,
5     contract_specification VARCHAR(150) NOT NULL,
6     reservation_id INT NOT NULL
7 );
8 ALTER TABLE contract
9 ADD CONSTRAINT fk_contract_reservation FOREIGN KEY (reservation_id) REFERENCES reservation(reservation_id);
```

## 9) create schema table

code.sql - SKYWAL\_KER001\proc (59)\* SQLQuery1.sql - SK...KER001\proc (69)\*

```

1 --schema table
2 -- View all tables in the current database
3 SELECT * FROM INFORMATION_SCHEMA.TABLES;
4
5 -- View columns for a specific table
6 SELECT * FROM INFORMATION_SCHEMA.COLUMNS
7
8 -- View foreign key constraints for a table
9 SELECT * FROM INFORMATION_SCHEMA.KEY_COLUMN_USAGE

```

76 %

Results [Messages]

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	TABLE_TYPE
1	laAtelier	dbo	insurance	BASE TABLE
2	laAtelier	dbo	employee	BASE TABLE
3	laAtelier	dbo	company	BASE TABLE
4	laAtelier	dbo	car	BASE TABLE
5	laAtelier	dbo	reservation	BASE TABLE
6	laAtelier	dbo	renter	BASE TABLE
7	laAtelier	dbo	payment	BASE TABLE
8	laAtelier	dbo	contract	BASE TABLE

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION	COLUMN_DEFAULT	IS_NULLABLE	DATA_TYPE	CHARACTER_MAXIMUM_LENGTH	CHARACTER_OCTET_LENGTH	NUMERIC_PRECISION	NUMERIC_PRECISION_RADIX	NUMERIC_SCALE	DATETIME_PRECIS
1	laAtelier	dbo	insurance	insurance_type	2	NULL	NO	varchar	50	50	10	10	0	NULL
2	laAtelier	dbo	insurance	insurance_com_	3	NULL	NO	varchar	150	150	NULL	NULL	NULL	NULL
3	laAtelier	dbo	insurance	company_email	4	NULL	NO	varchar	100	100	NULL	NULL	NULL	NULL
4	laAtelier	dbo	insurance	company_phon_	5	NULL	NO	varchar	20	20	NULL	NULL	NULL	NULL
5	laAtelier	dbo	employee	employee_id	1	NULL	NO	int	NULL	NULL	10	10	0	NULL
6	laAtelier	dbo	employee	first_name	2	NULL	NO	varchar	50	50	NULL	NULL	NULL	NULL
7	laAtelier	dbo	employee	last_name	3	NULL	NO	varchar	50	50	NULL	NULL	NULL	NULL

	CONSTRAINT_CATALOG	CONSTRAINT_SCHEMA	CONSTRAINT_NAME	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME	ORDINAL_POSITION
1	laAtelier	dbo	fk_car_id	laAtelier	dbo	reservation	car_id	1
2	laAtelier	dbo	fk_contract_reservation	laAtelier	dbo	contract	reservation_id	1
3	laAtelier	dbo	fk_employee_id	laAtelier	dbo	company	employee_id	1
4	laAtelier	dbo	fk_insurance_no	laAtelier	dbo	car	insurance_no	1
5	laAtelier	dbo	fk_owner_id	laAtelier	dbo	car	employee_id	1
6	laAtelier	dbo	fk_payment_reservation	laAtelier	dbo	payment	reservation_id	1
7	laAtelier	dbo	fk_reservation_id	laAtelier	dbo	renter	reservation_id	1
8	laAtelier	dbo	PK_car_4DAB0D	laAtelier	dbo	car	car_id	1
9	laAtelier	dbo	PK_company_3E2	laAtelier	dbo	company	company_id	1
10	laAtelier	dbo	PK_contact_F8D6	laAtelier	dbo	contract	contact_id	1
11	laAtelier	dbo	PK_employee_C5	laAtelier	dbo	employee	employee_id	1
12	laAtelier	dbo	PK_insurance_59A	laAtelier	dbo	insurance	insurance_no	1
13	laAtelier	dbo	PK_payment_ED1	laAtelier	dbo	payment	payment_id	1
14	laAtelier	dbo	PK_renter_5043F	laAtelier	dbo	renter	renter_id	1
15	laAtelier	dbo	PK_reservation_31B	laAtelier	dbo	reservation	reservation_id	1
16	laAtelier	dbo	UQ_car_97B9E5	laAtelier	dbo	car	number_plate	1

## Insert data into the database:

### 1. Populate insurance table

```
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
work_departments
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - insertingData.sql - LKER001\proce (69)
1 -- Populate insurance table
2 INSERT INTO insurance (insurance_no, insurance_type, insurance_company, company_email, company_phone_no)
3 VALUES
4 (10, 'Travel Insurance', 'JKL Insurance', 'travel@jklinsurance.com', '555-123-4567'),
5 (11, 'Pet Insurance', 'MNO Insurance', 'pet@mnoinsurance.com', '555-987-6543'),
6 (12, 'Rentalers Insurance', 'PQR Insurance', 'renters@pqrisurance.com', '555-222-3333'),
7 (13, 'Business Insurance', 'STU Insurance', 'business@stuinsurance.com', '555-444-5555'),
8 (14, 'Flood Insurance', 'VWX Insurance', 'flood@wxvinurance.com', '555-666-7777'),
9 (15, 'Health Insurance', 'ABC Health Insurance', 'health@abcinsurance.com', '555-111-2222'),
10 (16, 'Life Insurance', 'XYZ Life Insurance', 'life@xyzinsurance.com', '555-888-9999'),
11 (17, 'Home Insurance', 'DEF Home Insurance', 'home@definsurance.com', '555-333-4444'),
12 (18, 'Car Insurance', 'GHI Car Insurance', 'car@ghiinsurance.com', '555-555-6666'),
13 (19, 'Property Insurance', 'LMN Property Insurance', 'property@lmninsurance.com', '555-777-8888');
14 |
```

### 2. Populate employee table

```
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
work_departments
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - insertingData.sql - LKER001\proce (69)
1 -- Populate employee table
2 INSERT INTO employee (employee_id, first_name, last_name, phone_number, email, pay_rate)
3 VALUES
4 (17, 'Emma', 'Brown', '555-123-4567', 'emma.brown@example.com', 25000.00),
5 (18, 'Noah', 'Gonzalez', '555-234-5678', 'noah.gonzalez@example.com', 26000.00),
6 (19, 'Ava', 'Williams', '555-345-6789', 'ava.williams@example.com', 27000.00),
7 (20, 'Liam', 'Jones', '555-456-7890', 'liam.jones@example.com', 28000.00),
8 (21, 'Olivia', 'Garcia', '555-567-8901', 'olivia.garcia@example.com', 29000.00),
9 (22, 'William', 'Martinez', '555-678-9012', 'william.martinez@example.com', 30000.00),
10 (23, 'Sophia', 'Lee', '555-789-0123', 'sophia.lee@example.com', 31000.00),
11 (24, 'Mason', 'Hernandez', '555-890-1234', 'mason.hernandez@example.com', 32000.00),
12 (25, 'Amelia', 'Young', '555-901-2345', 'amelia.young@example.com', 33000.00),
13 (26, 'Logan', 'Scott', '555-012-3456', 'logan.scott@example.com', 34000.00);
14 |
```

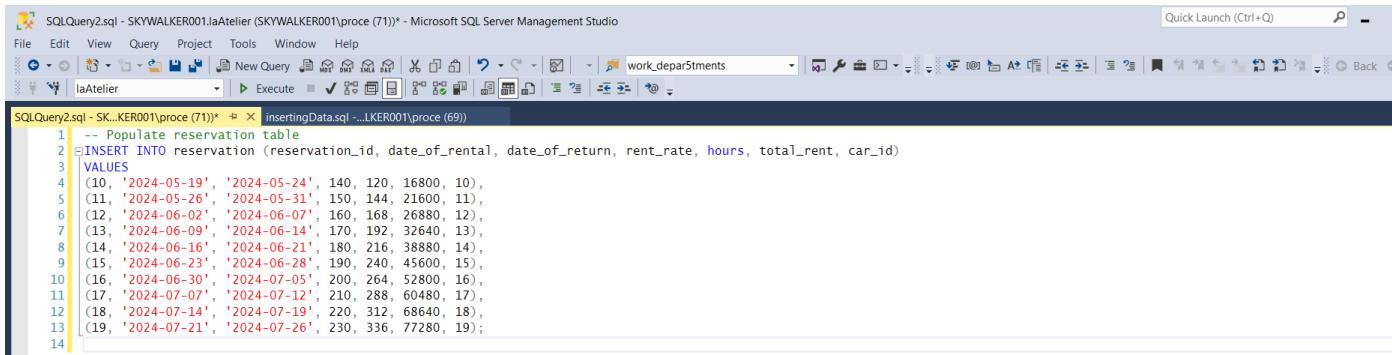
### 3. Populate company table

```
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
work_departments
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - insertingData.sql - LKER001\proce (69)
1 -- Populate company table
2 INSERT INTO company (company_id, company_name, phone_number, email, employee_id)
3 VALUES
4 (10, 'OPQ Enterprises', '555-111-2222', 'info@opqent.com', 17),
5 (11, 'RST Industries', '555-222-3333', 'info@rstind.com', 18),
6 (12, 'UVW Group', '555-333-4444', 'info@uvwgroup.com', 19),
7 (13, 'XYZ Services', '555-444-5555', 'info@xyzsvc.com', 20),
8 (14, 'LMN Corporation', '555-555-6666', 'info@lmncorp.com', 21),
9 (15, 'ABC Corporation', '555-666-7777', 'info@abccorp.com', 22),
10 (16, 'DEF Enterprises', '555-777-8888', 'info@defent.com', 23),
11 (17, 'GHI Industries', '555-888-9999', 'info@ghind.com', 24),
12 (18, 'JKL Group', '555-999-0000', 'info@jklgroup.com', 25),
13 (19, 'MNO Services', '555-000-1111', 'info@moscv.com', 26);|
```

### 4. Populate car table

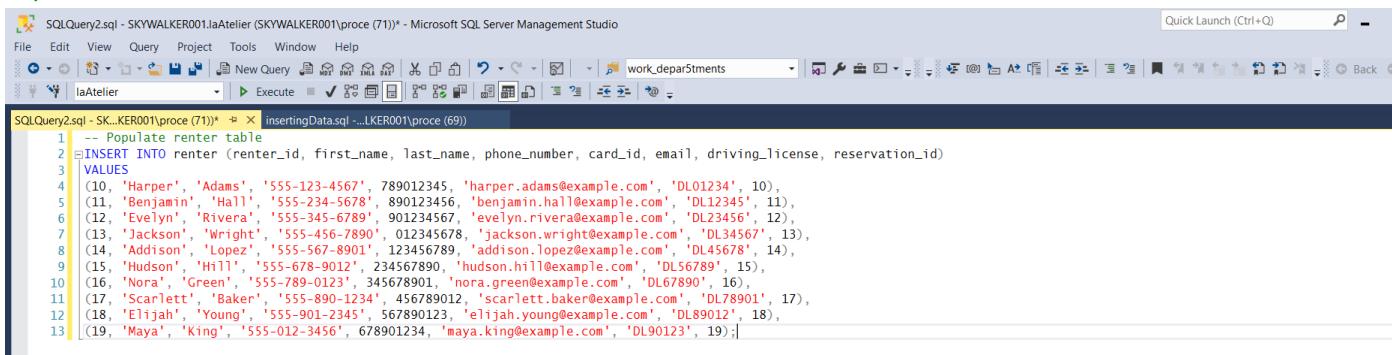
```
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - Microsoft SQL Server Management Studio
File Edit View Query Project Tools Window Help
work_departments
SQLQuery2.sql - SKYWALKER001\laAtelier (SKYWALKER001\proce (71)) - insertingData.sql - LKER001\proce (69)
1 -- Populate car table
2 INSERT INTO car (car_id, serial_no, car_model_name, number_plate, manufacturer, mfd, transmission, engine, tyres, rims, color, insurance_no, employee_id)
3 VALUES
4 (10, 'SNLMNP0', 'Toyota Prius', 'PRI123', 'Toyota', '2023-06-01', 'Automatic', '1.8L', 'Bridgestone', 'Alloy', 'Green', 10, 17),
5 (11, 'SNSTUVW', 'Honda Accord', 'ACC456', 'Honda', '2023-07-01', 'Automatic', '2.0L', 'Michelin', 'Alloy', 'Silver', 11, 18),
6 (12, 'SNXYZAB', 'Ford Fusion', 'FUS789', 'Ford', '2023-08-01', 'Automatic', '2.5L', 'Pirelli', 'Alloy', 'White', 12, 19),
7 (13, 'SNCDEFG', 'Chevrolet Malibu', 'MAL012', 'Chevrolet', '2023-09-01', 'Automatic', '2.5L', 'Goodyear', 'Alloy', 'Black', 13, 20),
8 (14, 'SNHJKL', 'Nissan Altima', 'ALT345', 'Nissan', '2023-10-01', 'Automatic', '2.5L', 'Firestone', 'Alloy', 'Red', 14, 21),
9 (15, 'SNMNOPE', 'Kia Optima', 'OPT678', 'Kia', '2023-11-01', 'Automatic', '2.4L', 'Bridgestone', 'Alloy', 'Blue', 15, 22),
10 (16, 'SNQRSTU', 'Hyundai Sonata', 'SON901', 'Hyundai', '2023-12-01', 'Automatic', '2.4L', 'Michelin', 'Alloy', 'Gray', 16, 23),
11 (17, 'SNEFGHI', 'Volkswagen Passat', 'PAS234', 'Volkswagen', '2024-01-01', 'Automatic', '2.0L', 'Goodyear', 'Alloy', 'Brown', 17, 24),
12 (18, 'SNJKLMN', 'Subaru Legacy', 'LEG567', 'Subaru', '2024-02-01', 'Automatic', '2.5L', 'Pirelli', 'Alloy', 'Gold', 18, 25),
13 (19, 'SNOPQRS', 'Mazda 6', 'MAZ890', 'Mazda', '2024-03-01', 'Automatic', '2.5L', 'Firestone', 'Alloy', 'Orange', 19, 26);|
```

## 5. Populate reservation table



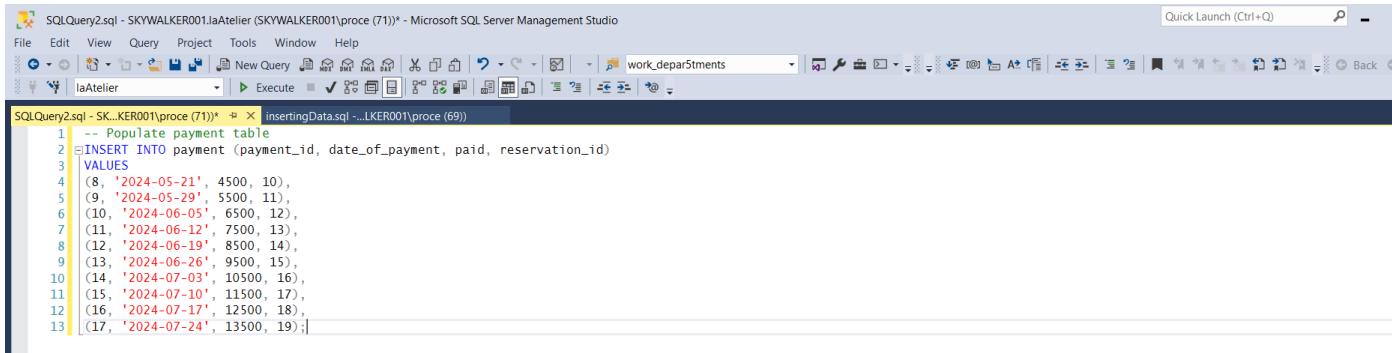
```
SQLQuery2.sql - SKYWALKER001\proce (71)* [ insertingData.sql - LKER001\proce (69) ]  
File Edit View Query Project Tools Window Help  
... work_departments ...  
laAtelier Execute Quick Launch (Ctrl+Q)  
  
1 -- Populate reservation table  
2 INSERT INTO reservation (reservation_id, date_of_rental, date_of_return, rent_rate, hours, total_rent, car_id)  
3 VALUES  
4 (10, '2024-05-19', '2024-05-24', 140, 120, 16800, 10),  
5 (11, '2024-05-26', '2024-05-31', 150, 144, 21600, 11),  
6 (12, '2024-06-02', '2024-06-07', 160, 168, 26880, 12),  
7 (13, '2024-06-09', '2024-06-14', 170, 192, 32640, 13),  
8 (14, '2024-06-16', '2024-06-21', 180, 216, 38880, 14),  
9 (15, '2024-06-23', '2024-06-28', 190, 240, 45600, 15),  
10 (16, '2024-06-30', '2024-07-05', 200, 264, 52800, 16),  
11 (17, '2024-07-07', '2024-07-12', 210, 288, 60480, 17),  
12 (18, '2024-07-14', '2024-07-19', 220, 312, 68640, 18),  
13 (19, '2024-07-21', '2024-07-26', 230, 336, 77280, 19);  
14
```

## 6. Populate renter table



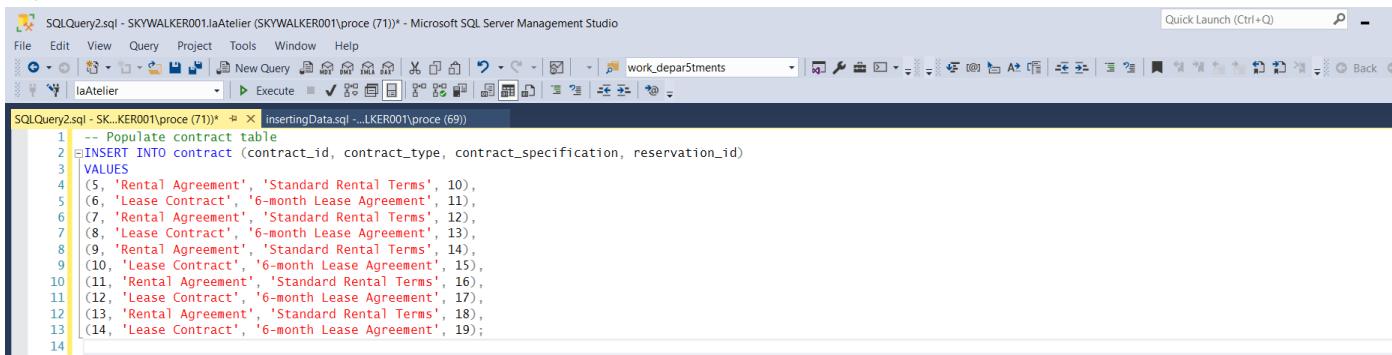
```
SQLQuery2.sql - SKYWALKER001\proce (71)* [ insertingData.sql - LKER001\proce (69) ]  
File Edit View Query Project Tools Window Help  
... work_departments ...  
laAtelier Execute Quick Launch (Ctrl+Q)  
  
1 -- Populate renter table  
2 INSERT INTO renter (renter_id, first_name, last_name, phone_number, card_id, email, driving_license, reservation_id)  
3 VALUES  
4 (10, 'Harper', 'Adams', '555-123-4567', 789012345, 'harper.adams@example.com', 'DL01234', 10),  
5 (11, 'Benjamin', 'Hall', '555-234-5678', 890123456, 'benjamin.hall@example.com', 'DL12345', 11),  
6 (12, 'Evelyn', 'Rivera', '555-345-6789', 901234567, 'evelyn.rivera@example.com', 'DL23456', 12),  
7 (13, 'Jackson', 'Wright', '555-456-7890', 012345678, 'jackson.wright@example.com', 'DL34567', 13),  
8 (14, 'Addison', 'Lopez', '555-567-8901', 123456789, 'addison.lopez@example.com', 'DL45678', 14),  
9 (15, 'Hudson', 'Hill', '555-678-9012', 234567890, 'hudson.hill@example.com', 'DL56789', 15),  
10 (16, 'Nora', 'Green', '555-789-0123', 345678901, 'nora.green@example.com', 'DL67890', 16),  
11 (17, 'Scarlett', 'Baker', '555-890-1234', 456789012, 'scarlett.baker@example.com', 'DL78901', 17),  
12 (18, 'Ellijah', 'Young', '555-901-2345', 567890123, 'ellijah.young@example.com', 'DL89012', 18),  
13 (19, 'Maya', 'King', '555-012-3456', 678901234, 'maya.king@example.com', 'DL90123', 19);  
14
```

## 7. Populate payment table



```
SQLQuery2.sql - SKYWALKER001\proce (71)* [ insertingData.sql - LKER001\proce (69) ]  
File Edit View Query Project Tools Window Help  
... work_departments ...  
laAtelier Execute Quick Launch (Ctrl+Q)  
  
1 -- Populate payment table  
2 INSERT INTO payment (payment_id, date_of_payment, paid, reservation_id)  
3 VALUES  
4 (8, '2024-05-21', 4500, 10),  
5 (9, '2024-05-29', 5500, 11),  
6 (10, '2024-06-05', 6500, 12),  
7 (11, '2024-06-12', 7500, 13),  
8 (12, '2024-06-19', 8500, 14),  
9 (13, '2024-06-26', 9500, 15),  
10 (14, '2024-07-03', 10500, 16),  
11 (15, '2024-07-10', 11500, 17),  
12 (16, '2024-07-17', 12500, 18),  
13 (17, '2024-07-24', 13500, 19);  
14
```

## 8. Populate contract table



```
SQLQuery2.sql - SKYWALKER001\proce (71)* [ insertingData.sql - LKER001\proce (69) ]  
File Edit View Query Project Tools Window Help  
... work_departments ...  
laAtelier Execute Quick Launch (Ctrl+Q)  
  
1 -- Populate contract table  
2 INSERT INTO contract (contract_id, contract_type, contract_specification, reservation_id)  
3 VALUES  
4 (5, 'Rental Agreement', 'Standard Rental Terms', 10),  
5 (6, 'Lease Contract', '6-month Lease Agreement', 11),  
6 (7, 'Rental Agreement', 'Standard Rental Terms', 12),  
7 (8, 'Lease Contract', '6-month Lease Agreement', 13),  
8 (9, 'Rental Agreement', 'Standard Rental Terms', 14),  
9 (10, 'Lease Contract', '6-month Lease Agreement', 15),  
10 (11, 'Rental Agreement', 'Standard Rental Terms', 16),  
11 (12, 'Lease Contract', '6-month Lease Agreement', 17),  
12 (13, 'Rental Agreement', 'Standard Rental Terms', 18),  
13 (14, 'Lease Contract', '6-month Lease Agreement', 19);  
14
```

## Constraints Testing:

### 1) Testing CHECK constraint on employee table:

The screenshot shows a SQL query window titled "constraintTesting..LKER001\proce (64)". The code attempts to insert a new employee record with a pay rate of 9000.00, which fails because it's less than the minimum value of 10000 defined by the CHECK constraint.

```
1 --Testing CHECK constraint on employee table:  
2 -- This should fail due to pay_rate less than 10000  
3 INSERT INTO employee (employee_id, first_name, last_name, phone_number, email, pay_rate)  
4 VALUES (27, 'John', 'Doe', '555-123-4567', 'john.doe@example.com', 9000.00);
```

Completion time: 2024-04-12T11:52:10.9432981-04:00

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_\_employee\_\_A1936A6B4CDBAA15'. Cannot insert duplicate key in object 'dbo.employee'. The duplicate key value is (555-123-4567).  
The statement has been terminated.

### 2) Testing UNIQUE constraint on employee table:

The screenshot shows a SQL query window titled "constraintTesting..LKER001\proce (64)". The code attempts to insert a new employee record with an email address that already exists in the database, failing due to the UNIQUE constraint.

```
1 --Testing CHECK constraint on employee table:  
2 -- This should fail due to email already existing  
3 INSERT INTO employee (employee_id, first_name, last_name, phone_number, email, pay_rate)  
4 VALUES (27, 'John', 'Doe', '555-123-4567', 'emma.brown@example.com', 10000.00);
```

Completion time: 2024-04-12T11:54:05.8289981-04:00

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_\_employee\_\_AB6E616495B8198D'. Cannot insert duplicate key in object 'dbo.employee'. The duplicate key value is (emma.brown@example.com).  
The statement has been terminated.

### 3) Testing FOREIGN KEY constraint on company table:

The screenshot shows a SQL query window titled "constraintTesting..LKER001\proce (64)". The code attempts to insert a new company record with an employee\_id that does not exist in the employee table, failing due to the FOREIGN KEY constraint.

```
1 --Testing CHECK constraint on employee table:  
2 -- This should fail due to non-existing employee_id  
3 INSERT INTO company (company_id, company_name, phone_number, email, employee_id)  
4 VALUES (20, 'Test Company', '555-123-4567', 'test@example.com', 100);
```

Completion time: 2024-04-12T11:55:03.8359089-04:00

Msg 547, Level 16, State 0, Line 3  
The INSERT statement conflicted with the FOREIGN KEY constraint "fk\_employee\_id". The conflict occurred in database "laAtelier", table "dbo.employee", column 'employee\_id'.  
The statement has been terminated.

#### 4) Testing FOREIGN KEY constraint on car table:

The screenshot shows a SQL query window in SSMS. The code attempts to insert a new row into the 'car' table with a non-existent insurance number:

```
--Testing FOREIGN KEY constraint on car table:  
-- This should fail due to non-existing insurance_no  
INSERT INTO car (car_id, serial_no, car_model_name, number_plate, manufacturer, mfd, transmission, engine, tyres, rims, color, insurance_no, employee_id)  
VALUES (20, 'SNTSTST', 'Test Car', 'TEST123', 'Test', '2024-04-01', 'Manual', '2.0L', 'Test Tyres', 'Test Rims',  
'Black', 100, 17);
```

The execution fails with the following error message in the 'Messages' pane:

Msg 547, Level 16, State 0, Line 3  
The INSERT statement conflicted with the FOREIGN KEY constraint "fk\_insurance\_no". The conflict occurred in database "laAtelier", table "dbo.insurance", column 'insurance\_no'.  
The statement has been terminated.

Completion time: 2024-04-12T11:56:04.8972771-04:00

#### 5) Testing FOREIGN KEY constraint on reservation table::

The screenshot shows a SQL query window in SSMS. The code attempts to insert a new row into the 'reservation' table with a non-existent car ID:

```
--Testing FOREIGN KEY constraint on reservation table:  
-- This should fail due to non-existing car_id  
INSERT INTO reservation (reservation_id, date_of_rental, date_of_return, rent_rate, hours, total_rent, car_id)  
VALUES (20, '2024-05-19', '2024-05-24', 140, 120, 16800, 100);
```

The execution fails with the following error message in the 'Messages' pane:

Msg 547, Level 16, State 0, Line 3  
The INSERT statement conflicted with the FOREIGN KEY constraint "fk\_car\_id". The conflict occurred in database "laAtelier", table "dbo.car", column 'car\_id'.  
The statement has been terminated.

Completion time: 2024-04-12T11:56:26.8931315-04:00

#### 6) Testing FOREIGN KEY constraint on renter table:

The screenshot shows a SQL query window in SSMS. The code attempts to insert a new row into the 'renter' table, which violates a unique key constraint:

```
--Testing FOREIGN KEY constraint on renter table:  
-- This should fail due to non-existing reservation_id  
INSERT INTO renter (renter_id, first_name, last_name, phone_number, card_id, email, driving_license, reservation_id)  
VALUES (20, 'Test', 'Renter', '555-123-4567', 123456789, 'test.renter@example.com', 'DL67890', 100);
```

The execution fails with the following error message in the 'Messages' pane:

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_\_renter\_\_CF3263C4A1C2A920'. Cannot insert duplicate key in object 'dbo.renter'. The duplicate key value is (DL67890).  
The statement has been terminated.

Completion time: 2024-04-12T12:24:53.6633260-04:00

## 7) Testing FOREIGN KEY constraint on payment table:

```
SQLQuery4.sql - SK...KER001\proce (73)* constraintTesting.s...LKER001\proce (64)
1 --Testing FOREIGN KEY constraint on payment table:
2 -- This should fail due to non-existing reservation_id
3 INSERT INTO payment (payment_id, date_of_payment, paid, reservation_id)
4 VALUES (18, '2024-07-24', 13500, 100);

136 % ▾
Messages
Msg 547, Level 16, State 0, Line 3
The INSERT statement conflicted with the FOREIGN KEY constraint "fk_payment_reservation". The conflict occurred in database "laAtelier", table "dbo.reservation", column 'reservation_id'.
The statement has been terminated.

Completion time: 2024-04-12T11:57:16.2286237-04:00
```

## 8) Testing FOREIGN KEY constraint on contract table

```
SQLQuery4.sql - SK...KER001\proce (73)* constraintTesting.s...LKER001\proce (64)
1 --Testing FOREIGN KEY constraint on contract table:
2 -- This should fail due to non-existing reservation_id
3 INSERT INTO contract (contract_id, contract_type, contract_specification, reservation_id)
4 VALUES (15, 'Test Contract', 'Test Specification', 100);

136 % ▾
Messages
Msg 547, Level 16, State 0, Line 3
The INSERT statement conflicted with the FOREIGN KEY constraint "fk_contract_reservation". The conflict occurred in database "laAtelier", table "dbo.reservation", column 'reservation_id'.
The statement has been terminated.

Completion time: 2024-04-12T11:57:50.3103811-04:00
```

## 9) Testing UNIQUE constraint on insurance table:

```
SQLQuery4.sql - SK...KER001\proce (54)* constraintTesting.sql - not connected createDatabase.sql - not connected*
1 --9--Testing UNIQUE constraint on insurance table:
2 -- This should fail due to duplicate insurance_no
3 INSERT INTO insurance (insurance_no, insurance_type, insurance_company, company_email, company_phone_no)
4 VALUES (10, 'Test Insurance', 'Test Company', 'test@example.com', '555-123-4567');

113 % ▾
Messages
Msg 2627, Level 14, State 1, Line 3
Violation of PRIMARY KEY constraint 'PK_insurance_58AC8992E10298ED'. Cannot insert duplicate key in object 'dbo.insurance'. The duplicate key value is (10).
The statement has been terminated.

Completion time: 2024-04-12T14:13:18.7166302-04:00
```

## 10) Testing FOREIGN KEY constraint on car table for employee\_id:

```
SQLQuery4.sql - SK...KER001\proce (54)* constraintTesting.sql - not connected createDatabase.sql - not connected*
1 --10--Testing FOREIGN KEY constraint on car table for employee_id:
2 -- This should fail due to non-existing employee_id
3 INSERT INTO car (car_id, serial_no, car_model_name, number_plate, manufacturer, mfd, transmission, engine, tyres, rims, color, insurance_no, employee_id)
4 VALUES (21, 'SNTSTST', 'Test Car', 'TEST123', 'Test', '2024-04-01', 'Manual', '2.0L', 'Test Tyres', 'Test Rims', 'Black', 10, 100);
5

113 % ▾
Messages
Msg 547, Level 16, State 0, Line 3
The INSERT statement conflicted with the FOREIGN KEY constraint "fk_owner_id". The conflict occurred in database "laAtelier", table "dbo.employee", column 'employee_id'.
The statement has been terminated.

Completion time: 2024-04-12T14:13:37.6628971-04:00
```

## 11) Testing FOREIGN KEY constraint on renter table for card\_id:

The screenshot shows a SQL Server Management Studio window with the following details:

- Query Editor: SQLQuery4.sql - SK\_KER001\proce (54)\*
- Results: constraintTesting.sql - not connected
- Script: createDatabase.sql - not connected\*
- Text:

```
1 --11--Testing FOREIGN KEY constraint on renter table for card_id
2 -- This should fail due to non-existing card_id
3 INSERT INTO renter (renter_id, first_name, last_name, phone_number, card_id, email, driving_license, reservation_id)
4 VALUES (21, 'Test', 'Renter', '555-123-4567', 999999999, 'test.renter@example.com', 'DL67890', 10);
```
- Messages:

```
Msg 2627, Level 14, State 1, Line 3
Violation of UNIQUE KEY constraint 'UQ__renter__CF3263C4A1C2A920'. Cannot insert duplicate key in object 'dbo.renter'. The duplicate key value is (DL67890).
The statement has been terminated.
```
- Completion time: 2024-04-12T14:13:55.4357877-04:00

## 12) Testing NOT NULL constraint on various tables:

The screenshot shows a SQL Server Management Studio window with the following details:

- Query Editor: SQLQuery4.sql - SK\_KER001\proce (54)\*
- Results: constraintTesting.sql - not connected
- Script: createDatabase.sql - not connected\*
- Text:

```
1 --12--Testing NOT NULL constraint on various tables:
2 -- This should fail due to NULL value in required column 'rent_rate' in reservation table
3 INSERT INTO reservation (reservation_id, date_of_rental, date_of_return, hours, total_rent, car_id)
4 VALUES (20, '2024-05-19', '2024-05-24', NULL, 16800, 10);
```
- Messages:

```
Msg 515, Level 16, State 2, Line 3
Cannot insert the value NULL into column 'hours', table 'laAtelier.dbo.reservation'; column does not allow nulls. INSERT fails.
The statement has been terminated.
```
- Completion time: 2024-04-12T14:14:14.1572876-04:00

## 13) Testing CHECK constraint on payment table:

The screenshot shows a SQL Server Management Studio window with the following details:

- Query Editor: SQLQuery4.sql - SK\_KER001\proce (54)\*
- Results: constraintTesting.sql - not connected
- Script: createDatabase.sql - not connected\*
- Text:

```
1 --13--Testing CHECK constraint on payment table:
2 -- This should fail due to negative 'paid' value
3 INSERT INTO payment (payment_id, date_of_payment, paid, reservation_id)
4 VALUES (18, '2024-07-24', -500, 10);
```
- Messages:

```
Msg 2627, Level 14, State 1, Line 3
Violation of PRIMARY KEY constraint 'PK__payment__ED1FC9EADCC1BB0D'. Cannot insert duplicate key in object 'dbo.payment'. The duplicate key value is (18).
The statement has been terminated.
```
- Completion time: 2024-04-12T14:14:27.4997258-04:00

## 14) Testing UNIQUE constraint on company table for email:

The screenshot shows a SQL Server Management Studio window with the following details:

- Query Editor: createDatabase.sql - KFR001\proce (55)
- Results: SQLQuery4.sql - SK\_KER001\proce (68)\*
- Script: constraintTesting.sql - not connected
- Text:

```
1 --15--Testing FOREIGN KEY constraint on car table for insurance_no:
2 -- This should fail due to non-existing insurance_no
3 INSERT INTO car (car_id, serial_no, car_model_name, number_plate, manufacturer, mfd, transmission, engine, tyres, ~
4 rims, color, insurance_no, employee_id)
5 VALUES (21, 'SNTSTST', 'Test Car', 'TEST123', 'Test', '2024-04-01', 'Manual', '2.0L', 'Test Tyres', 'Test Rims', ~
6 'Black', 100, 17);
```
- Messages:

```
Msg 547, Level 16, State 0, Line 3
The INSERT statement conflicted with the FOREIGN KEY constraint "fk_insurance_no". The conflict occurred in database "laAtelier", table "dbo.insurance", column 'insurance_no'.
The statement has been terminated.
```
- Completion time: 2024-04-12T13:02:28.8311178-04:00

### 15) Testing FOREIGN KEY constraint on car table for insurance\_no:

The screenshot shows a SQL Server Management Studio window with two tabs: 'createDatabase.sql...KER001\proce (55)' and 'SQLQuery4.sql - SK...KER001\proce (68)\*'. The second tab is active and contains the following SQL code:

```
1 --14--Testing UNIQUE constraint on company table for email:  
2 -- This should fail due to duplicate email  
3 INSERT INTO company (company_id, company_name, phone_number, email, employee_id)  
4 VALUES (20, 'Test Company', '555-123-4567', 'info@opqent.com', 100);
```

In the 'Messages' pane at the bottom, an error message is displayed:

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_\_company\_\_AB6E616483C5EE56'. Cannot insert duplicate key in object 'dbo.company'. The duplicate key value is (info@opqent.com).  
The statement has been terminated.

Completion time: 2024-04-12T13:01:42.2007945-04:00

### 16) Testing FOREIGN KEY constraint on contract table for reservation\_id:

The screenshot shows a SQL Server Management Studio window with two tabs: 'createDatabase.sql...KER001\proce (55)' and 'SQLQuery4.sql - SK...KER001\proce (68)\*'. The second tab is active and contains the following SQL code:

```
1 --16--Testing FOREIGN KEY constraint on contract table for reservation_id:  
2 -- This should fail due to non-existing reservation_id  
3 INSERT INTO contract (contract_id, contract_type, contract_specification, reservation_id)  
4 VALUES (15, 'Test Contract', 'Test Specification', 100);
```

In the 'Messages' pane at the bottom, an error message is displayed:

Msg 547, Level 16, State 0, Line 3  
The INSERT statement conflicted with the FOREIGN KEY constraint "fk\_contract\_reservation". The conflict occurred in database "laAtelier", table "dbo.reservation", column 'reservation\_id'.  
The statement has been terminated.

Completion time: 2024-04-12T13:02:44.4327849-04:00

### 17) Testing NOT NULL constraint on renter table for driving\_license:

The screenshot shows a SQL Server Management Studio window with two tabs: 'createDatabase.sql...KER001\proce (55)' and 'SQLQuery4.sql - SK...KER001\proce (68)\*'. The second tab is active and contains the following SQL code:

```
1 --17--Testing NOT NULL constraint on renter table for driving_license:  
2 -- This should fail due to NULL value in required column 'driving_license'  
3 INSERT INTO renter (renter_id, first_name, last_name, phone_number, card_id, email, reservation_id)  
4 VALUES (21, 'Test', 'Renter', '555-123-4567', 999999999, 'test.renter@example.com', 10);
```

In the 'Messages' pane at the bottom, an error message is displayed:

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_\_renter\_\_A1936A6B260D98E9'. Cannot insert duplicate key in object 'dbo.renter'. The duplicate key value is (555-123-4567).  
The statement has been terminated.

Completion time: 2024-04-12T13:03:20.3602207-04:00

### 18) Testing NOT NULL constraint on reservation table for car\_id:

The screenshot shows a SQL Server Management Studio window with two tabs: 'createDatabase.sql...KER001\proce (55)\*' and 'SQLQuery4.sql - SK...KER001\proce (68)\*'. The second tab is active and contains the following SQL code:

```
1 --19--Testing CHECK constraint on employee table for pay_rate:  
2 -- This should fail due to pay_rate less than 10000  
3 INSERT INTO employee (employee_id, first_name, last_name, phone_number, email, pay_rate)  
4 VALUES (27, 'John', 'Doe', '555-123-4567', 'john.doe@example.com', 9000.00);
```

In the 'Messages' pane at the bottom, an error message is displayed:

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_\_employee\_\_A1936A6B4CDBAA15'. Cannot insert duplicate key in object 'dbo.employee'. The duplicate key value is (555-123-4567).  
The statement has been terminated.

Completion time: 2024-04-12T13:20:44.1375393-04:00

## 19) Testing CHECK constraint on employee table for pay\_rate:

```
createDatabase.sql...KER001\proce (55)* SQLQuery4.sql - SK...KER001\proce (68)* constraintTesting.sql - not connected
1 --18--Testing NOT NULL constraint on reservation table for car_id
2 -- This should fail due to NULL value in required column 'car_id'
3 INSERT INTO reservation (reservation_id, date_of_rental, date_of_return, rent_rate, hours, total_rent, car_id)
4 VALUES (20, '2024-05-19', '2024-05-24', 140, 120, 16800, NULL);
5
6
```

136 %

Messages

```
Msg 515, Level 16, State 2, Line 3
Cannot insert the value NULL into column 'car_id', table 'laAtelier.dbo.reservation'; column does not allow nulls. INSERT fails.
The statement has been terminated.
```

Completion time: 2024-04-12T13:11:18.2750525-04:00

## 20) Testing FOREIGN KEY constraint on car table for employee\_id:

```
createDatabase.sql...KER001\proce (55)* SQLQuery4.sql - SK...KER001\proce (68)* constraintTesting.sql - not connected
1
2 --20--Testing FOREIGN KEY constraint on car table for employee_id:
3 -- This should fail due to non-existing employee_id
4 INSERT INTO car (car_id, serial_no, car_model_name, number_plate, manufacturer, mfd, transmission, engine, tyres, rims, color,
5 insurance_no, employee_id)
6 VALUES (21, 'SNTSTST', 'Test Car', 'TEST123', 'Test', '2024-04-01', 'Manual', '2.0L', 'Test Tyres', 'Test Rims', 'Black', 10,
7 100);
```

136 %

Messages

```
Msg 547, Level 16, State 0, Line 4
The INSERT statement conflicted with the FOREIGN KEY constraint "fk_owner_id". The conflict occurred in database "laAtelier", table "dbo.employee", column 'employee_id'.
The statement has been terminated.
```

Completion time: 2024-04-12T13:21:08.0355435-04:00

## 21) Testing FOREIGN KEY constraint on renter table for card\_id:

```
createDatabase.sql - not connected* SQLQuery4.sql - SK...KER001\proce (54)* constraintTesting.sql - not connected
1 --21--Testing FOREIGN KEY constraint on renter table for card_id:
2 -- This should fail due to non-existing card_id
3 INSERT INTO renter (renter_id, first_name, last_name, phone_number, card_id, email, driving_license, reservation_id)
4 VALUES (21, 'Test', 'Renter', '555-123-4567', 999999999, 'test.renter@example.com', 'DL67890', 10);
```

136 %

Messages

```
Msg 2627, Level 14, State 1, Line 3
Violation of UNIQUE KEY constraint 'UQ__renter__CF3263C4A1C2A920'. Cannot insert duplicate key in object 'dbo.renter'. The duplicate key value is (DL67890).
The statement has been terminated.
```

Completion time: 2024-04-12T13:50:12.0605222-04:00

## 22) Testing CHECK constraint on payment table for paid:

```
createDatabase.sql - not connected* SQLQuery4.sql - not connected* constraintTesting.sql - not connected
1 --22--Testing CHECK constraint on payment table for paid:
2 -- This should fail due to negative 'paid' value
3 INSERT INTO payment (payment_id, date_of_payment, paid, reservation_id)
4 VALUES (18, '2024-07-24', -500, 10);
```

136 %

Messages

```
Msg 102, Level 15, State 1, Line 3
The INSERT statement conflicted with the CHECK constraint "chk_paid_non_negative". The conflict occurred in database "your_database_name", table "dbo.payment", column 'paid'.
The statement has been terminated.
```

Completion time: 2024-04-12T13:21:24.9781719-04:00

## 23) Testing CHECK constraint on reservation table for hours:

```
createDatabase.sql - not connected* SQLQuery4.sql - SK...KER001\proce (54)* constraintTesting.sql - not connected
1 --23--Testing CHECK constraint on reservation table for hours:
2 -- This should fail due to negative 'hours' value
3 INSERT INTO reservation (reservation_id, date_of_rental, date_of_return, rent_rate, hours, total_rent, car_id)
4 VALUES (20, '2024-05-19', '2024-05-24', 140, -120, 16800, 10);
```

113 %

Messages

```
Msg 2627, Level 14, State 1, Line 3
The INSERT statement conflicted with the CHECK constraint "CK_reservation_hours_xxx". The conflict occurred in database "your_database_name", table "dbo.reservation", column 'hours'.
The statement has been terminated.
```

Completion time: 2024-04-12T13:52:22.9509085-04:00

#### 24) Testing CHECK constraint on insurance table for insurance\_no:

The screenshot shows a SQL Server Management Studio window with two tabs: 'createDatabase.sql - not connected\*' and 'constraintTesting.sql - not connected'. The 'constraintTesting.sql' tab contains the following SQL code:

```
1 --24--Testing CHECK constraint on insurance table for insurance_no;
2 -- This should fail due to duplicate insurance_no
3 INSERT INTO insurance (insurance_no, insurance_type, insurance_company, company_email, company_phone_no)
4 VALUES (10, 'Test Insurance', 'Test Company', 'test@example.com', '555-123-4567');
```

In the 'Messages' pane, an error message is displayed:

Msg 2627, Level 14, State 1, Line 3  
Violation of PRIMARY KEY constraint 'PK\_insurance\_58AC8992E1029EDE'. Cannot insert duplicate key in object 'dbo.insurance'. The duplicate key value is (10).  
The statement has been terminated.

Completion time: 2024-04-12T14:00:04.8300758-04:00

#### 25) Testing CHECK constraint on employee table for pay\_rate:

The screenshot shows a SQL Server Management Studio window with two tabs: 'createDatabase.sql - not connected\*' and 'constraintTesting.sql - not connected'. The 'constraintTesting.sql' tab contains the following SQL code:

```
1 --25--Testing CHECK constraint on employee table for pay_rate;
2 -- This should fail due to pay_rate less than 10000
3 INSERT INTO employee (employee_id, first_name, last_name, phone_number, email, pay_rate)
4 VALUES (27, 'John', 'Doe', '555-123-4567', 'john.doe@example.com', 9000.00);
```

In the 'Messages' pane, an error message is displayed:

Msg 2627, Level 14, State 1, Line 3  
Violation of UNIQUE KEY constraint 'UQ\_employee\_A1996A6B4CDBAA15'. Cannot insert duplicate key in object 'dbo.employee'. The duplicate key value is (555-123-4567).  
The statement has been terminated.

Completion time: 2024-04-12T14:00:21.1918264-04:00

## Other Queries:

1)

SQLQuery6.sql - SK...KER001\proce (63)\* → queries.sql - SKW...KER001\proce (64)\*

```
1 -- Query to calculate the total rent paid for the month of June
2 SELECT SUM(total_rent) AS total_rent_paid_in_june -- Calculate the sum of total_rent column and alias it as
   total_rent_paid_in_june
3 FROM reservation -- Select data from the reservation table
4 WHERE MONTH(date_of_rental) = 6; -- Filter the reservations for the month of June
5
```

Results Messages

total_rent_paid_in_june
196800

Template Browser

- SQL Server Templates
  - Aggregate
  - Assembly
  - Audit
  - Backup
  - Certificate
  - Change Data Capture
  - Change Tracking
  - Credential
  - Database
  - Database Mail
  - Database Role
  - Database Trigger
  - Default
  - Earlier Versions
  - Endpoint
  - Extended Property
  - External Data Source
  - External File Format
  - Full-text
  - Function

2)

SQLQuery6.sql - SK...KER001\proce (63)\* → queries.sql - SKW...KER001\proce (64)\*

```
1 -- Query to calculate the average rent rate across all reservations
2 SELECT AVG(rent_rate) AS average_rent_rate -- Calculate the average of rent_rate column and alias it as average_rent_rate
3 FROM reservation; -- Select data from the reservation table
4
```

Results Messages

average_rent_rate
180

Template Browser

- SQL Server Templates
  - Aggregate
  - Assembly
  - Audit
  - Backup
  - Certificate
  - Change Data Capture
  - Change Tracking
  - Credential
  - Database
  - Database Mail
  - Database Role
  - Database Trigger
  - Default
  - Earlier Versions
  - Endpoint
  - Extended Property
  - External Data Source
  - External File Format
  - Full-text
  - Function

3)

queries.sql - SKW...KER001\proce (64)\* →

```
1 -- Query to select renter_email and phone_number for reservations where the payment amount is greater than 10000
2 SELECT renter_email, renter.phone_number
3 FROM renter
4 JOIN payment ON renter.reservation_id = payment.reservation_id
5 WHERE payment.paid > 10000;
```

Results Messages

email	phone_number
nora.green@example.com	555-789-0123
scarlett.baker@example.com	555-990-1234
elijah.young@example.com	555-901-2345
maya.king@example.com	555-012-3456

4)

queries.sql - SKW...KER001\proce (64)\* →

```
1 -- Query to select the date of rental, total rent, and number of reservations for each date
2 SELECT date_of_rental, SUM(total_rent) AS total_rent, COUNT(reservation_id) AS reservation_count
3 FROM reservation
4 GROUP BY date_of_rental;
5
```

Results Messages

date_of_rental	total_rent	reservation_count
2024-05-19	33600	2
2024-05-26	21600	1
2024-06-02	26880	1
2024-06-09	32640	1
2024-06-16	38880	1
2024-06-23	45600	1
2024-06-30	52800	1
2024-07-07	60480	1
2024-07-14	68640	1
2024-07-21	77280	1

5)

queries.sql - SKYW...KER001\proce (64)\*

```

1 -- Query to select car details for reservations made by a renter named 'Emma'
2 SELECT c.* 
3 FROM car c
4 JOIN reservation res ON c.car_id = res.car_id
5 JOIN renter r ON r.reservation_id = res.reservation_id
6 WHERE r.first_name = 'Jackson';

```

Results Messages

car_id	serial_no	car_model_name	number_plate	manufacturer	mfd	transmission	engine	tyres	rims	color	insurance_no	employee_id
13	SNCDEFG	Chevrolet Malibu	MAL012	Chevrolet	2023-09-01	Automatic	2.5L	Goodyear	Alloy	Black	13	20

6)

queries.sql - SKYW...KER001\proce (64)\*

```

1 -- Create a view to select silver cars from the car table
2 CREATE VIEW SilverCarsView AS
3 SELECT car_id, serial_no, car_model_name, number_plate
4 FROM car
5 WHERE color = 'Silver';
6
7 -- Query to select data from the SilverCarsView
8 SELECT * FROM SilverCarsView;
9

```

Results Messages

car_id	serial_no	car_model_name	number_plate
11	SNSTUVW	Honda Accord	ACC456

7)

queries.sql - SKYW...KER001\proce (64)\*

```

1 -- Query to select reservations with a rent rate greater than 150
2 SELECT * FROM reservation WHERE rent_rate > 150;

```

Results Messages

reservation_id	date_of_rental	date_of_return	rent_rate	hours	total_rent	car_id
12	2024-06-02	2024-06-07	160	168	26880	12
13	2024-06-09	2024-06-14	170	192	32640	13
14	2024-06-16	2024-06-21	180	216	38880	14
15	2024-06-23	2024-06-28	190	240	45600	15
16	2024-06-30	2024-07-05	200	264	52800	16
17	2024-07-07	2024-07-12	210	288	60480	17
18	2024-07-14	2024-07-19	220	312	68640	18
19	2024-07-21	2024-07-26	230	336	77280	19

8)

queries.sql - SKYW...KER001\proce (64)\*

```

1 -- Query to view to combine rental information for cars rented in May and July
2 -- This includes car ID, car model, car number plate, renter ID, renter name, and rental date
3 SELECT car.car_id, car.car_model_name, car.number_plate, renter.renter_id, CONCAT(renter.first_name, ' ', renter.last_name) AS renter_name,
4 reservation.date_of_rental
5 FROM reservation
6 JOIN car ON reservation.car_id = car.car_id
7 JOIN renter ON reservation.reservation_id = renter.reservation_id
8 WHERE MONTH(reservation.date_of_rental) = 5 -- May
9
10 UNION ALL
11
12 SELECT car.car_id, car.car_model_name, car.number_plate, renter.renter_id, CONCAT(renter.first_name, ' ', renter.last_name) AS renter_name,
13 reservation.date_of_rental
14 FROM reservation
15 JOIN car ON reservation.car_id = car.car_id
16 JOIN renter ON reservation.reservation_id = renter.reservation_id
17 WHERE MONTH(reservation.date_of_rental) = 7; -- July

```

Results Messages

car_id	car_model_name	number_plate	renter_id	renter_name	date_of_rental
10	Toyota Prius	PR123	10	Harper Adams	2024-05-19
11	Honda Accord	ACC456	11	Benjamin Hall	2024-05-26
17	Volkswagen Passat	PAS234	17	Scarlett Baker	2024-07-07
18	Subaru Legacy	LEG567	18	Elijah Young	2024-07-14
19	Mazda 6	MAZ890	19	Maya King	2024-07-21

9)

queries.sql - SKYW...KER001\proce (64)\*

```

1 -- Query to select all reservations and order them by date of rental
2 SELECT * FROM reservation ORDER BY date_of_rental;
3

```

113 %

Results Messages

	reservation_id	date_of_rental	date_of_return	rent_rate	hours	total_rent	car_id
1	10	2024-05-19	2024-05-24	140	120	16800	10
2	20	2024-05-19	2024-05-24	140	-120	16800	10
3	11	2024-05-26	2024-05-31	150	144	21600	11
4	12	2024-06-02	2024-06-07	160	168	26880	12
5	13	2024-06-08	2024-06-14	170	192	32640	13
6	14	2024-06-16	2024-06-21	180	216	38880	14
7	15	2024-06-23	2024-06-28	190	240	45600	15
8	16	2024-06-30	2024-07-05	200	264	52800	16
9	17	2024-07-07	2024-07-12	210	288	60480	17
10	18	2024-07-14	2024-07-19	220	312	68640	18
11	19	2024-07-21	2024-07-26	230	336	77280	19

10)

queries.sql - SKYW...KER001\proce (64)\*

```

1 -- Query to select the employees with the highest pay rate
2 SELECT first_name, last_name, pay_rate
3 FROM employee
4 WHERE pay_rate = (SELECT MAX(pay_rate) FROM employee);

```

93 %

Results Messages

	first_name	last_name	pay_rate
1	Logan	Scott	34000.00