A
Project Report
on

CAR RENTAL MANAGEMENT SYSTEM

Guided by
Internal Guide:
Prof. Fatema Vhora
Department of Information Technology
Faculty of Technology
DD University

Developed by

Mufaddal Suratwala(IT-164) – Department of IT, DD University
Raj Vachhani(IT-172) - Department of IT, DD University
Arshad Vhora(IT-178) - Department of IT, DD University

Department of Information Technology Faculty of Technology, Dharmsinh Desai University
College Road, Nadiad-387001
2021

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

DHARMSINH DESAI UNIVERSITY

NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled "**Car Rental Management System**" is a bonafied report of the work carried out by

1) **Mr. Mufaddal J. Suratwala,** Student ID No : **19ITUOS119**
2) **Mr. Raj Vachhani** , Student ID No : **19ITUOS087**
3) **Mr. Arshad S. Vhora** , Student ID No : **19ITUOS111** of Department of

Information Technology, semester V, under the guidance and supervision for the subject Database Management System. They were involved in Project training during academic year 2019-2020.

Prof. Fatema Vhora

(Project Guide)
Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

Prof. Vipul Dabhi

Head , Department of Information Technology,
Faculty of Technology,
Dharmsinh Desai University, Nadiad
Date:

ACKNOWLEDGEMENT

We would like to give our sincere acknowledgement to everybody responsible for the successful completion of our project “CAR RENTAL MANAGEMENT SYSTEM”.

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of this project.

We owe our deep gratitude to our project guide Prof. Fatema Vhora, who took keen interest on our project work and guided us all along till the completion of our project work by providing all the necessary help for developing an efficient Database System.

We would also like to thank all our lecturers.

Finally, we convoy our acknowledgement to all our friends and family members who directly or indirectly associated with us in the successful completion of the project. We thank one and all.

TABLE OF CONTENTS

1. SYSTEM OVERVIEW

1.1 CURRENT SYSTEM

Car Rental Management System has been designed to bridge the process of car rental to 3rd party customers from 3rd party owners for a fixed number of days.

Indians are becoming part of growing digital India. Since all rental systems are moving online we have tried to bring about a transparent system to ensure ease of services to both parties. The current system does not ensure the authenticity of the renter's license and the authenticity of the vehicle's information. Through our database, we will ensure that the rentee has all documents needed to drive the vehicle and the renter has his vehicle in proper condition with all safety features of the vehicle known to the rentee.

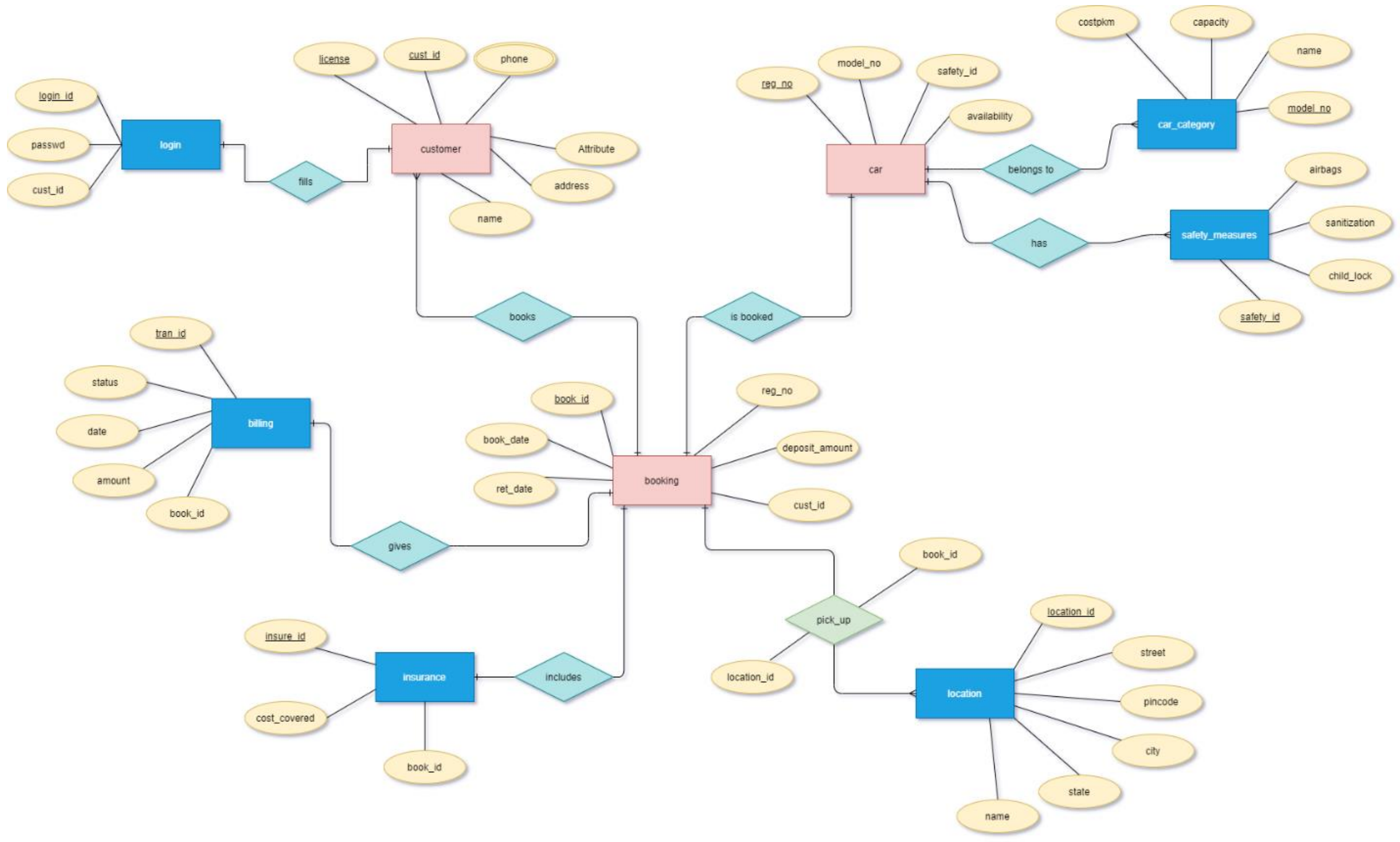
1.2 OBJECTIVES OF THE PROPOSED SYSTEM

- It aims to provide authentic information about both parties before rental and tries to filter the customer needs using its unique filter system to meet the needs of a customer without much trouble.
- Providing an efficient medium for smooth transactions between the customer and the renter by keeping an up-to-date database about both parties.
- Trying to reach different parts of the country and meet the needs of the customers in the swiftest way possible.

1.3 ADVANTAGES OF THE PROPOSED) SYSTEM

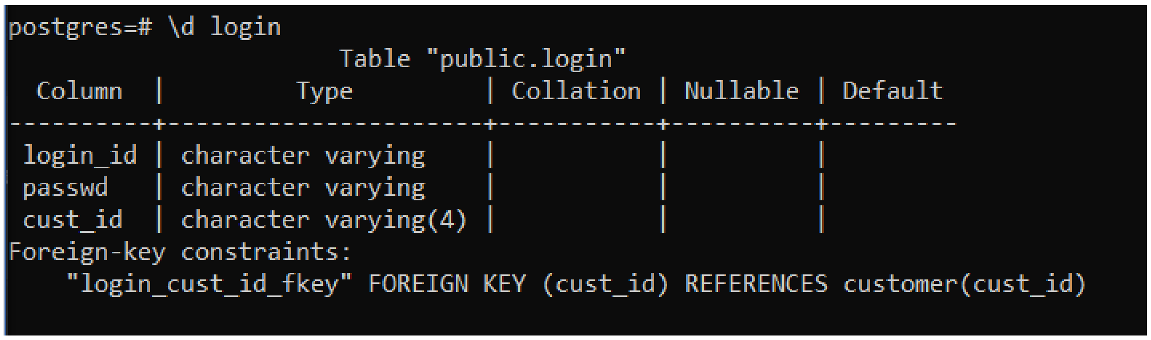
- It allows 3rd party customers to rent cars from 3rd party owners.
- It allows customers to rent cars on an online platform for a fixed no. of days.
- It helps filter the vehicle according to the customer's needs.
- The system provides detailed information about the car's safety features to the customer.
- It assures the owner about the rentee's authenticity.
- It provides insurance to the customer in case the vehicle malfunctions
- It provides an accurate pick-up location to the customer.
- It helps provide transparency in the transaction between the customer and the owner.

2. E-R DIAGRAM


login_id
passwd
cust_id
login
fills
license
cust_id
phone
customer
Attribute
address
name
reg_no
model_no
safety_id
availability
car
belongs to
costpkm
capacity
name
model_no
car_category
has
airbags
sanitization
child_lock
safety_id
safety_measures
books
is booked
tran_id
status
date
amount
book_id
billing
book_id
book_date
ret_date
booking
reg_no
deposit_amount
cust_id
gives
book_id
pick_up
location_id
location_id
street
pincode
city
state
name
location
insure_id
cost_covered
book_id
insurance
includes

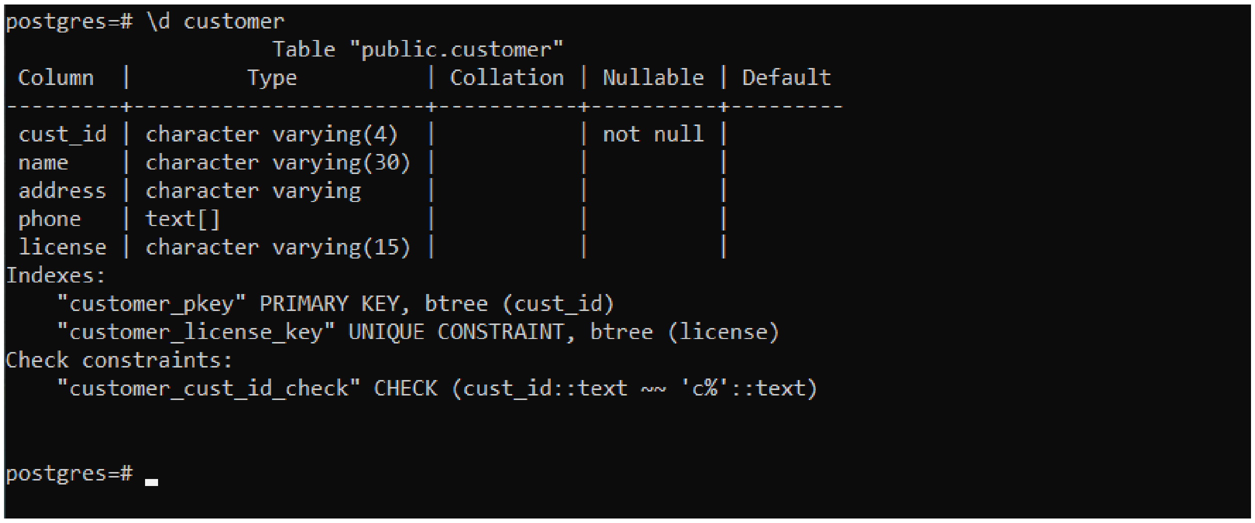
3. DATA DICTIONARY

3.1 Login



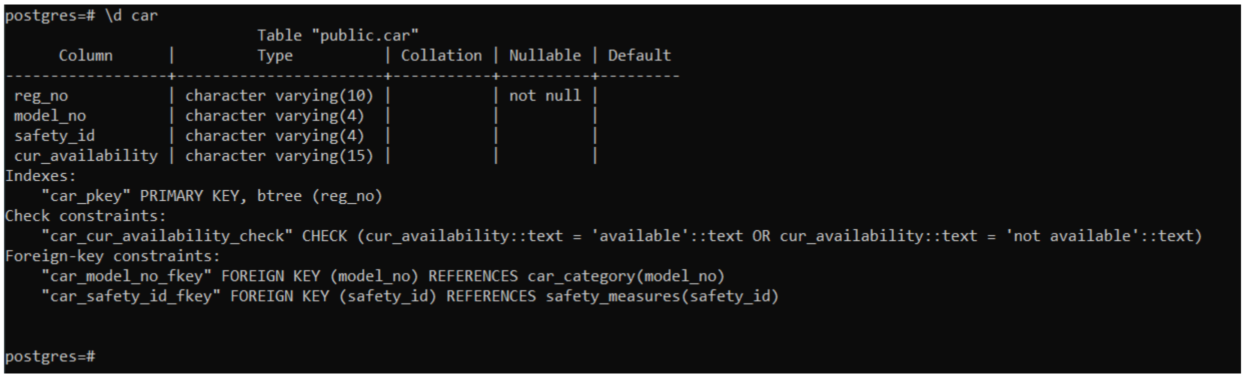
```
postgres=# \d login
                       Table "public.login"
  Column   |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 login_id | character varying     |           |          |
 passwd   | character varying     |           |          |
 cust_id  | character varying(4) |           |          |
Foreign-key constraints:
    "login_cust_id_fkey" FOREIGN KEY (cust_id) REFERENCES customer(cust_id)
```

3.2 Customer



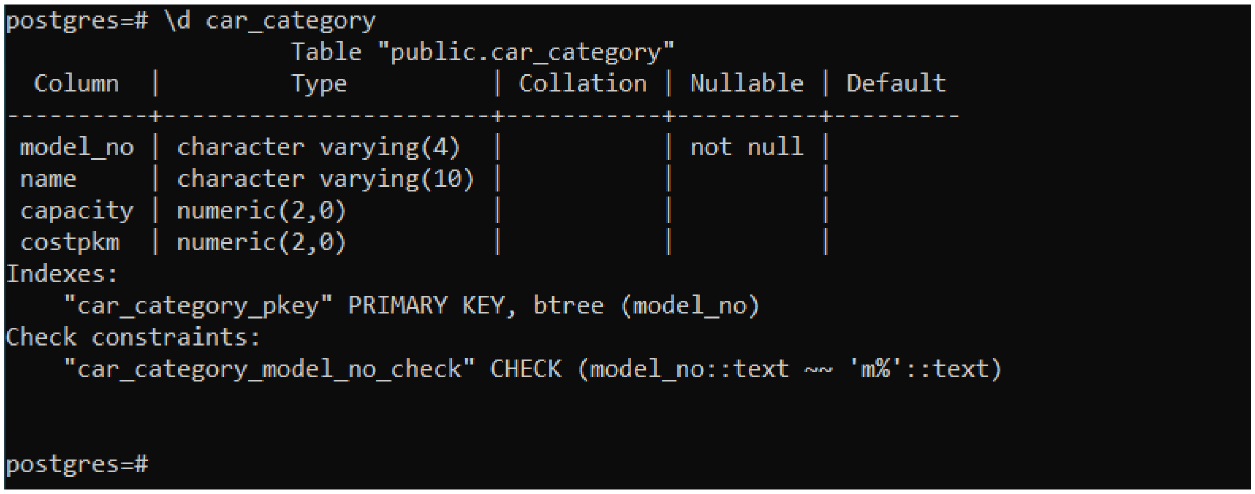
```
postgres=# \d customer
                     Table "public.customer"
 Column  |         Type          | Collation | Nullable | Default
---------+-----------------------+-----------+----------+---------
 cust_id | character varying(4)  |           | not null |
 name    | character varying(30) |           |          |
 address | character varying     |           |          |
 phone   | text[]                |           |          |
 license | character varying(15) |           |          |
Indexes:
    "customer_pkey" PRIMARY KEY, btree (cust_id)
    "customer_license_key" UNIQUE CONSTRAINT, btree (license)
Check constraints:
    "customer_cust_id_check" CHECK (cust_id::text ~~ 'c%'::text)


postgres=#
```

3.3 Car



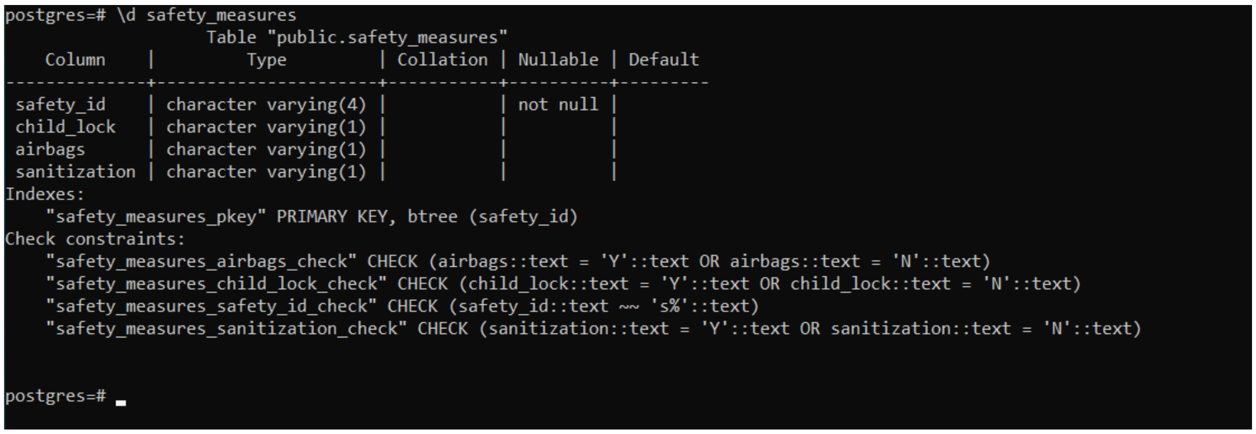
```
postgres=# \d car
                          Table "public.car"
      Column      |         Type          | Collation | Nullable | Default
------------------+-----------------------+-----------+----------+---------
 reg_no           | character varying(10) |           | not null |
 model_no         | character varying(4)  |           |          |
 safety_id        | character varying(4)  |           |          |
 cur_availability | character varying(15) |           |          |
Indexes:
    "car_pkey" PRIMARY KEY, btree (reg_no)
Check constraints:
    "car_cur_availability_check" CHECK (cur_availability::text = 'available'::text OR cur_availability::text = 'not available'::text)
Foreign-key constraints:
    "car_model_no_fkey" FOREIGN KEY (model_no) REFERENCES car_category(model_no)
    "car_safety_id_fkey" FOREIGN KEY (safety_id) REFERENCES safety_measures(safety_id)


postgres=#
```

3.4 Car Category



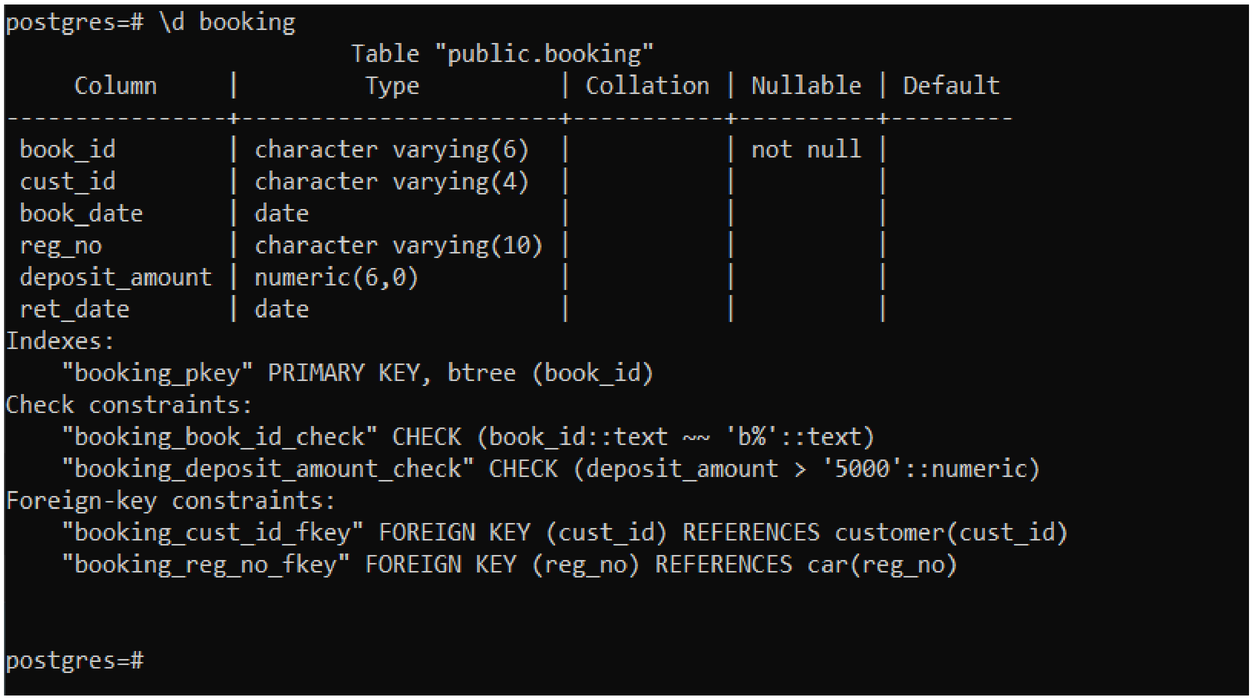
```
postgres=# \d car_category
                    Table "public.car_category"
  Column  |         Type          | Collation | Nullable | Default
----------+-----------------------+-----------+----------+---------
 model_no | character varying(4)  |           | not null |
 name     | character varying(10) |           |          |
 capacity | numeric(2,0)          |           |          |
 costpkm  | numeric(2,0)          |           |          |
Indexes:
    "car_category_pkey" PRIMARY KEY, btree (model_no)
Check constraints:
    "car_category_model_no_check" CHECK (model_no::text ~~ 'm%'::text)


postgres=#
```

3.5 Safety Measures



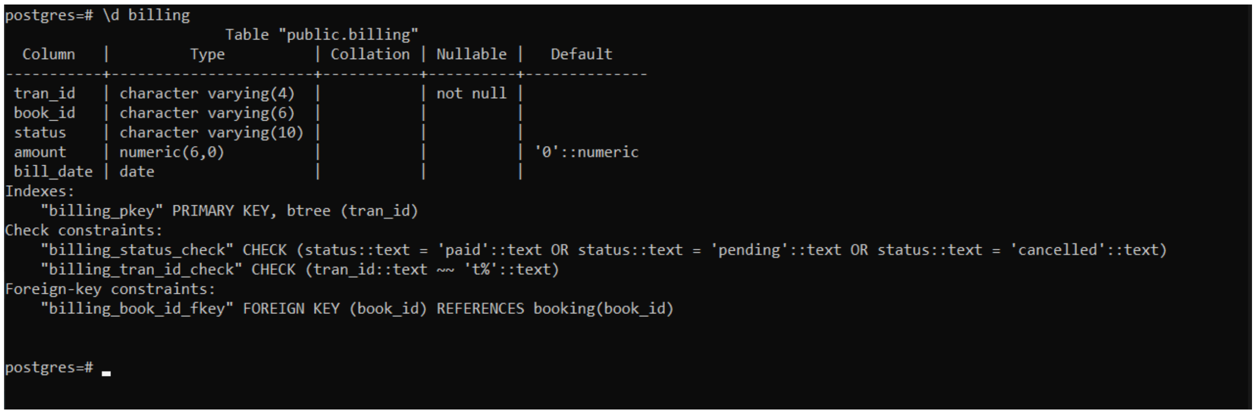
```
postgres=# \d safety_measures
                    Table "public.safety_measures"
    Column    |         Type         | Collation | Nullable | Default
--------------+----------------------+-----------+----------+---------
 safety_id    | character varying(4) |           | not null |
 child_lock   | character varying(1) |           |          |
 airbags      | character varying(1) |           |          |
 sanitization | character varying(1) |           |          |
Indexes:
    "safety_measures_pkey" PRIMARY KEY, btree (safety_id)
Check constraints:
    "safety_measures_airbags_check" CHECK (airbags::text = 'Y'::text OR airbags::text = 'N'::text)
    "safety_measures_child_lock_check" CHECK (child_lock::text = 'Y'::text OR child_lock::text = 'N'::text)
    "safety_measures_safety_id_check" CHECK (safety_id::text ~~ 's%'::text)
    "safety_measures_sanitization_check" CHECK (sanitization::text = 'Y'::text OR sanitization::text = 'N'::text)


postgres=#
```

3.6 Booking



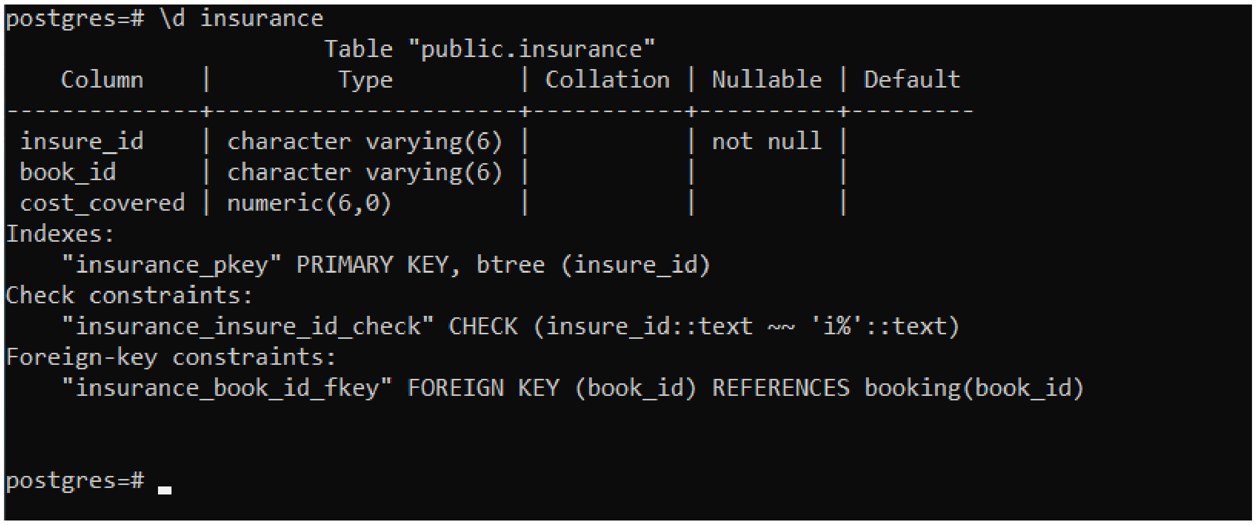
```
postgres=# \d booking
                         Table "public.booking"
     Column     |          Type          | Collation | Nullable | Default
----------------+------------------------+-----------+----------+---------
 book_id        | character varying(6)  |           | not null |
 cust_id        | character varying(4)  |           |          |
 book_date      | date                   |           |          |
 reg_no         | character varying(10) |           |          |
 deposit_amount | numeric(6,0)           |           |          |
 ret_date       | date                   |           |          |
Indexes:
    "booking_pkey" PRIMARY KEY, btree (book_id)
Check constraints:
    "booking_book_id_check" CHECK (book_id::text ~~ 'b%'::text)
    "booking_deposit_amount_check" CHECK (deposit_amount > '5000'::numeric)
Foreign-key constraints:
    "booking_cust_id_fkey" FOREIGN KEY (cust_id) REFERENCES customer(cust_id)
    "booking_reg_no_fkey" FOREIGN KEY (reg_no) REFERENCES car(reg_no)


postgres=#
```

3.7 Billing



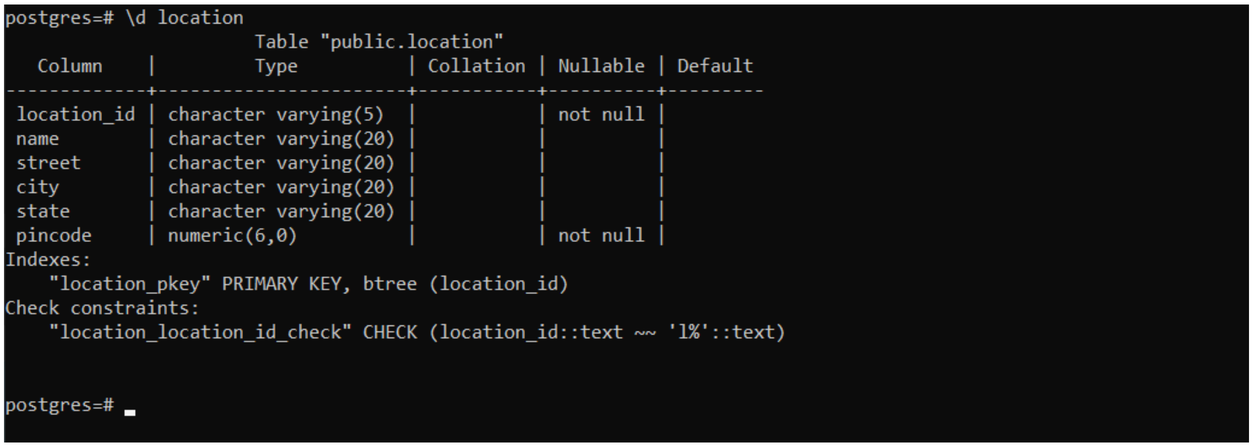
```
postgres=# \d billing
                         Table "public.billing"
  Column   |         Type          | Collation | Nullable |   Default
-----------+-----------------------+-----------+----------+--------------
 tran_id   | character varying(4)  |           | not null |
 book_id   | character varying(6)  |           |          |
 status    | character varying(10) |           |          |
 amount    | numeric(6,0)          |           |          | '0'::numeric
 bill_date | date                  |           |          |
Indexes:
    "billing_pkey" PRIMARY KEY, btree (tran_id)
Check constraints:
    "billing_status_check" CHECK (status::text = 'paid'::text OR status::text = 'pending'::text OR status::text = 'cancelled'::text)
    "billing_tran_id_check" CHECK (tran_id::text ~~ 't%'::text)
Foreign-key constraints:
    "billing_book_id_fkey" FOREIGN KEY (book_id) REFERENCES booking(book_id)


postgres=#
```

3.8 Insurance



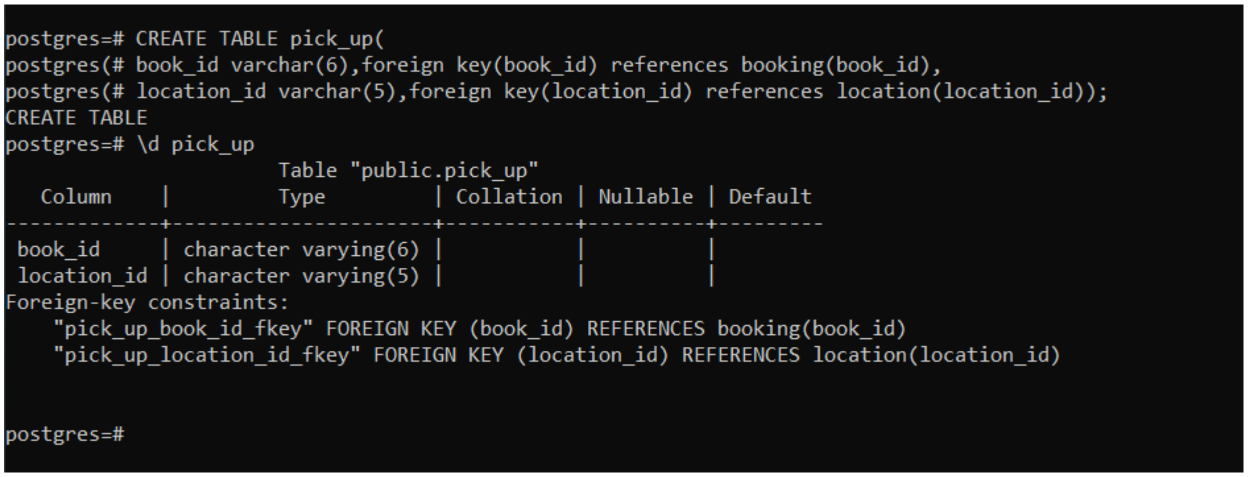
```
postgres=# \d insurance
                       Table "public.insurance"
    Column    |         Type         | Collation | Nullable | Default
--------------+----------------------+-----------+----------+---------
 insure_id    | character varying(6) |           | not null |
 book_id      | character varying(6) |           |          |
 cost_covered | numeric(6,0)         |           |          |
Indexes:
    "insurance_pkey" PRIMARY KEY, btree (insure_id)
Check constraints:
    "insurance_insure_id_check" CHECK (insure_id::text ~~ 'i%'::text)
Foreign-key constraints:
    "insurance_book_id_fkey" FOREIGN KEY (book_id) REFERENCES booking(book_id)


postgres=#
```

3.9 Location



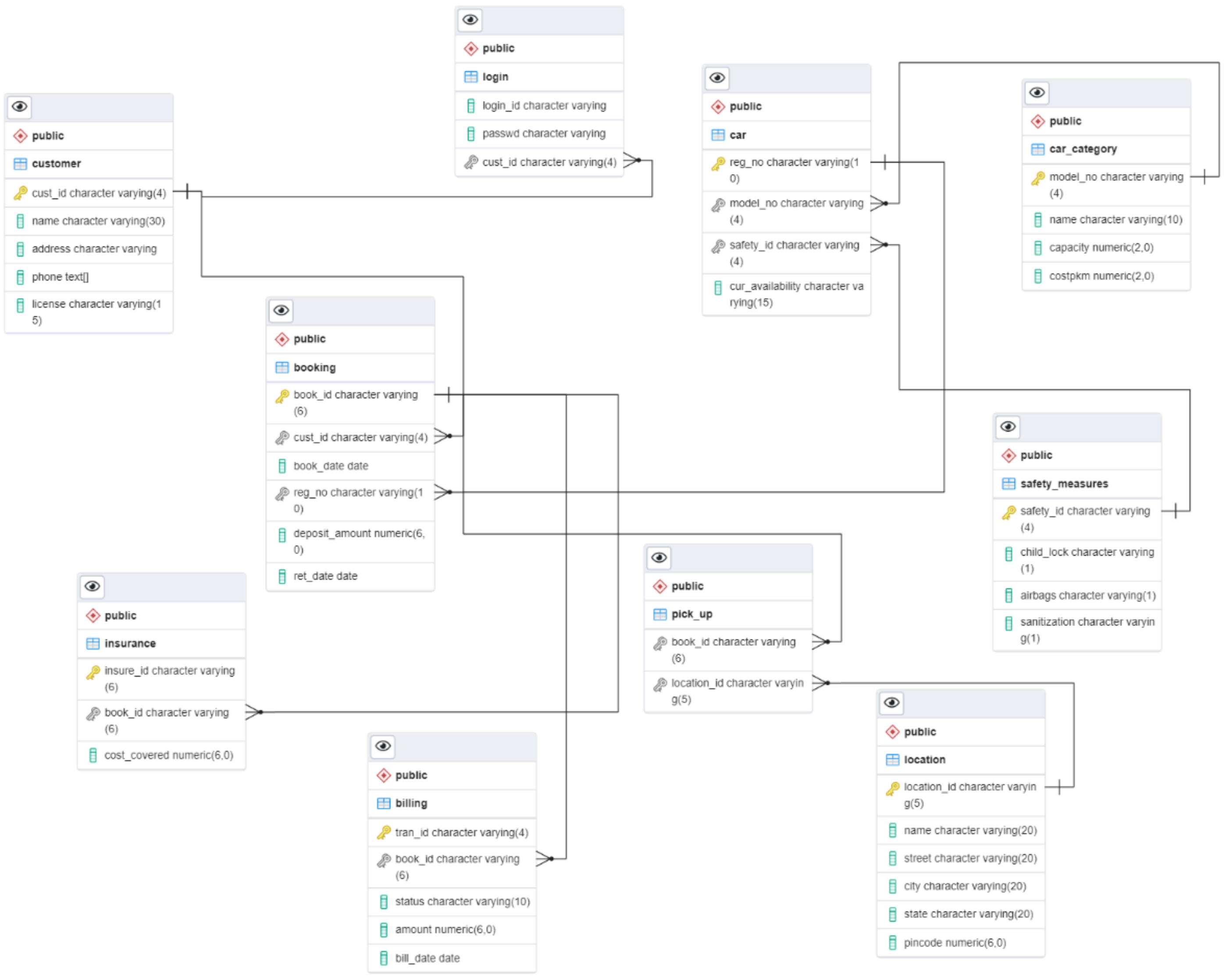
```
postgres=# \d location
                       Table "public.location"
   Column    |         Type          | Collation | Nullable | Default
-------------+-----------------------+-----------+----------+---------
 location_id | character varying(5)  |           | not null |
 name        | character varying(20) |           |          |
 street      | character varying(20) |           |          |
 city        | character varying(20) |           |          |
 state       | character varying(20) |           |          |
 pincode     | numeric(6,0)          |           | not null |
Indexes:
    "location_pkey" PRIMARY KEY, btree (location_id)
Check constraints:
    "location_location_id_check" CHECK (location_id::text ~~ 'l%'::text)


postgres=#
```

3.10 Pick-up



```
postgres=# CREATE TABLE pick_up(
postgres(# book_id varchar(6),foreign key(book_id) references booking(book_id),
postgres(# location_id varchar(5),foreign key(location_id) references location(location_id));
CREATE TABLE
postgres=# \d pick_up
                      Table "public.pick_up"
   Column    |         Type         | Collation | Nullable | Default
-------------+----------------------+-----------+----------+---------
 book_id     | character varying(6) |           |          |
 location_id | character varying(5) |           |          |
Foreign-key constraints:
    "pick_up_book_id_fkey" FOREIGN KEY (book_id) REFERENCES booking(book_id)
    "pick_up_location_id_fkey" FOREIGN KEY (location_id) REFERENCES location(location_id)


postgres=#
```

4. SCHEMA DIAGRAM


public
login
login_id character varying
passwd character varying
cust_id character varying(4)
public
customer
cust_id character varying(4)
name character varying(30)
address character varying
phone text[]
license character varying(15)
public
car
reg_no character varying(10)
model_no character varying(4)
safety_id character varying(4)
cur_availability character varying(15)
public
car_category
model_no character varying(4)
name character varying(10)
capacity numeric(2,0)
costpkm numeric(2,0)
public
booking
book_id character varying(6)
cust_id character varying(4)
book_date date
reg_no character varying(10)
deposit_amount numeric(6,0)
ret_date date
public
safety_measures
safety_id character varying(4)
child_lock character varying(1)
airbags character varying(1)
sanitization character varying(1)
public
insurance
insure_id character varying(6)
book_id character varying(6)
cost_covered numeric(6,0)
public
pick_up
book_id character varying(6)
location_id character varying(5)
public
billing
tran_id character varying(4)
book_id character varying(6)
status character varying(10)
amount numeric(6,0)
bill_date date
public
location
location_id character varying(5)
name character varying(20)
street character varying(20)
city character varying(20)
state character varying(20)
pincode numeric(6,0)

4. DATABASE IMPLEMENTATION

5.1 CREATE SCHEMA

5.1.1 Login

```
postgres=# CREATE TABLE login (
postgres(#  login_id varchar,
postgres(#  passwd varchar,
postgres(#  cust_id varchar(4), foreign key(cust_id) references customer(cust_id));
CREATE TABLE
```

5.1.2 Customer

```
postgres=# CREATE TABLE customer (
postgres(#  cust_id varchar(4) check(cust_id like 'c%') primary key,
postgres(#  name varchar(30),
postgres(#  address varchar,
postgres(#  phone text[],
postgres(#  license varchar(15) unique);
CREATE TABLE
```

5.1.3 Car

```
postgres=#  CREATE TABLE car (
postgres(#  reg_no varchar(10) primary key,
postgres(# model_no varchar(4),foreign key(model_no) references
car_category(model_no),
postgres(# safety_id varchar(4),foreign key(safety_id) references
safety_measures(safety_id),
postgres(# cur_availability varchar(15) CHECK(cur_availability=’available’ or
cur_availability=’not available’));
CREATE TABLE
```

5.1.4 Car Category

```
postgres=# CREATE TABLE car_category (
postgres(# model_no varchar(4) primary key CHECK (model_no like 'm%'),
postgres(# name varchar(10),
postgres(# capacity numeric(2),
postgres(# costpkm numeric(2));
CREATE TABLE
```

5.1.5 Safety Measures

```
postgres=# CREATE TABLE safety_measures (
postgres(# safety_id varchar(4) primary key CHECK (safety_id like 's%'),
postgres(# child_lock varchar(1) CHECK (child_lock='Y' or child_lock='N'),
postgres(# airbags varchar(1) CHECK (airbags='Y' or airbags='N'),
postgres(# sanitization varchar(1) CHECK (sanitization='Y' or sanitization='N'));
CREATE TABLE
```

5.1.6 Booking

```
postgres=# CREATE TABLE booking (
postgres(# book_id varchar(6) primary key check(book_id like 'b%'),
postgres(# cust_id varchar(4),foreign key(cust_id) references customer(cust_id),
postgres(# book_date date,
postgres(# reg_no varchar(10),foreign key(reg_no) references car(reg_no),
postgres(# deposit_amount numeric(6) check(deposit_amount>’5000’),
postgres(# ret_date date);
CREATE TABLE
```

5.1.7 Billing

```
postgres=# CREATE TABLE billing (
postgres(# tran_id varchar(4) primary key check (tran_id like 't%'),
postgres(#  book_id varchar(6),foreign key(book_id) references booking(book_id),
postgres(# status varchar(10) check(status='paid' or status='pending' or
status='cancelled'),
postgres(# amount numeric(6) DEFAULT '0',
postgres(#  bill_date date);
CREATE TABLE
```

5.1.8 Insurance

```
postgres=# CREATE TABLE insurance (
postgres(# insure_id varchar(6) primary key CHECK (insure_id like 'i%'),
postgres(# book_id varchar(6),foreign key(book_id) references booking(book_id),
postgres(# cost_covered numeric(6));
CREATE TABLE
```

5.1.9 Location

```
postgres=# CREATE TABLE location(
postgres(# location_id varchar(5) primary key CHECK (location_id like 'l%'),
postgres(# name varchar(20),
postgres(# street varchar(20),
postgres(# city varchar(20),
postgres(# state varchar(20),
postgres(# pincode numeric(6) not null);
CREATE TABLE
```

5.1.10 Pick-up

```
postgres=# CREATE TABLE pick_up(
postgres(# book_id varchar(6),foreign key(book_id) references booking(book_id),
postgres(# location_id varchar(5),foreign key(location_id) references
location(location_id));
CREATE TABLE
```

5.2 INSERT DATA VALUE

5.2.1 Login

INSERT INTO login values

('nd780289fd@tlead.me','hgvsdg7652','c101'),

('saadsaad12120@iprloi.com','jhfghee8753','c156'),

('ltehseen.gorsi1@drstshop.com','kjbjijmej76463','c165'),

('msa-2006-bek0@pdfrn.site','jh4534543hjg4','c201'),

('zmllell@azel.xyz','khnd37646jkb','c256'),

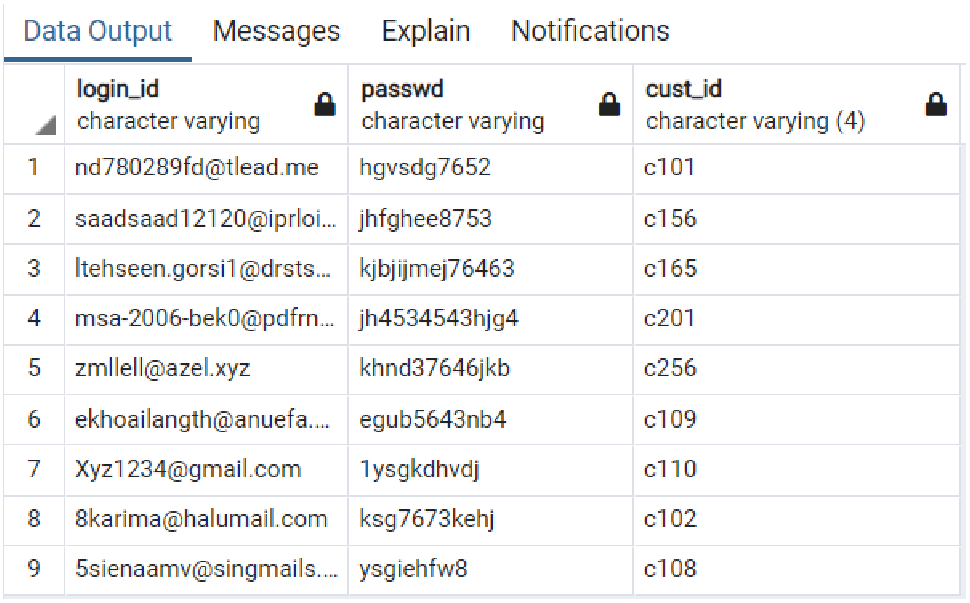
('ekhoailangth@anuefa.com','egub5643nb4','c109'),

('Xyz1234@gmail.com','1ysgkdhvdj','c110'),

('8karima@halumail.com','ksg7673kehj','c102'),

('5sienaamv@singmails.com','ysgiehfw8','c108');

INSERT 0 9



	login_id character varying	passwd character varying	cust_id character varying (4)
1	nd780289fd@tlead.me	hgvsdg7652	c101
2	saadsaad12120@iprloi...	jhfghee8753	c156
3	ltehseen.gorsi1@drsts...	kjbjijmej76463	c165
4	msa-2006-bek0@pdfrn...	jh4534543hjg4	c201
5	zmllell@azel.xyz	khnd37646jkb	c256
6	ekhoailangth@anuefa....	egub5643nb4	c109
7	Xyz1234@gmail.com	1ysgkdhvdj	c110
8	8karima@halumail.com	ksg7673kehj	c102
9	5sienaamv@singmails....	ysgiehfw8	c108

5.2.2 Customer

INSERT INTO customer values

('c101','Harsh Sanghvi','Umiyaji Krupa, street no 2, Navrang society, Rajkot, Gujarat',array['7844456789','4545454544'],'GJ0312345678910'),

('c156','Yash Patel','B101, Patel street, near market, Ring Road, Jamnagar, Gujarat',array['8765432198'],'GJ1075876543210'),

('c165','Jay Shah','55/d, Dhirendra Nath Ghosh Rd, Bhawanipore, Kolkata, West Bengal',array['6531652473','5454587899'],'WB0185674326541'),

('c201','Keval Desai','33, Dr Ambedkar Road, Parel, Mumbai, Maharashtra',array['7658543216'],'MH0467854367285'),

('c256','Rohit Sharma','22-5-227/a/g, Kothi, Hyderabad, Andhra Pradesh',array['6534231456'],'AP0164874532764'),

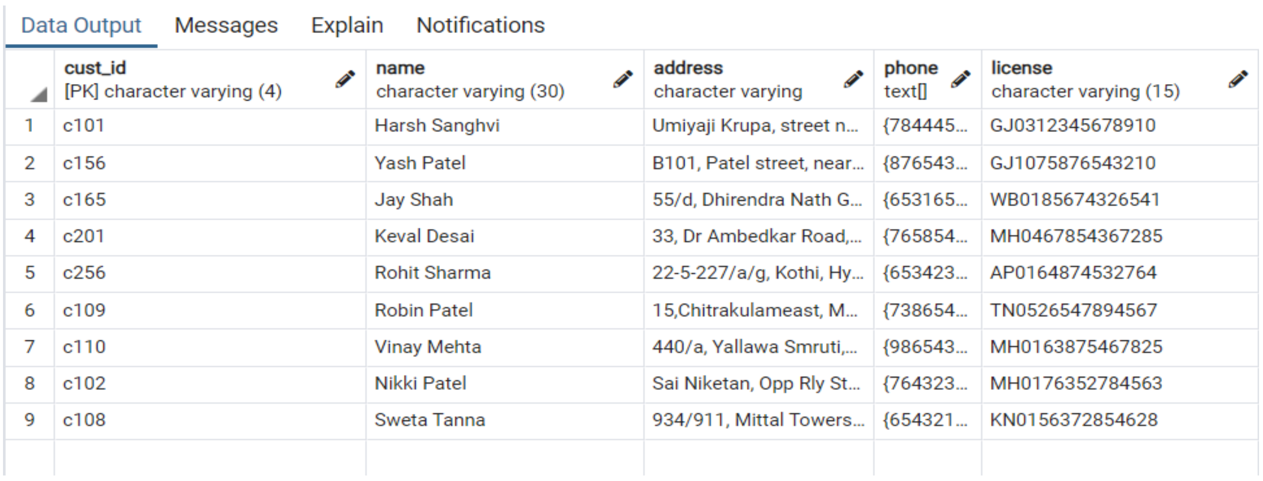
('c109','Robin Patel','15,Chitrakulameast, Mylapore,Chennai,Tamil Nadu',array['7386546565','5579897788'],'TN0526547894567'),

('c110','Vinay Mehta','440/a, Yallawa Smruti,Chagla Rd, Sahar, Andheri (west), Mumbai, Maharashtra',array['9865435687'],'MH0163875467825'),

('c102','Nikki Patel','Sai Niketan, Opp Rly Station, S V Road, Borivali (west), Mumbai, Maharashtra',array['7643234567'],'MH0176352784563'),

('c108','Sweta Tanna','934/911, Mittal Towers, 9th Floor,b Wing M G Road, M G Road, Bangalore, Karnataka',array['6543216789','8787998989'],'KN0156372854628');

INSERT 0 9

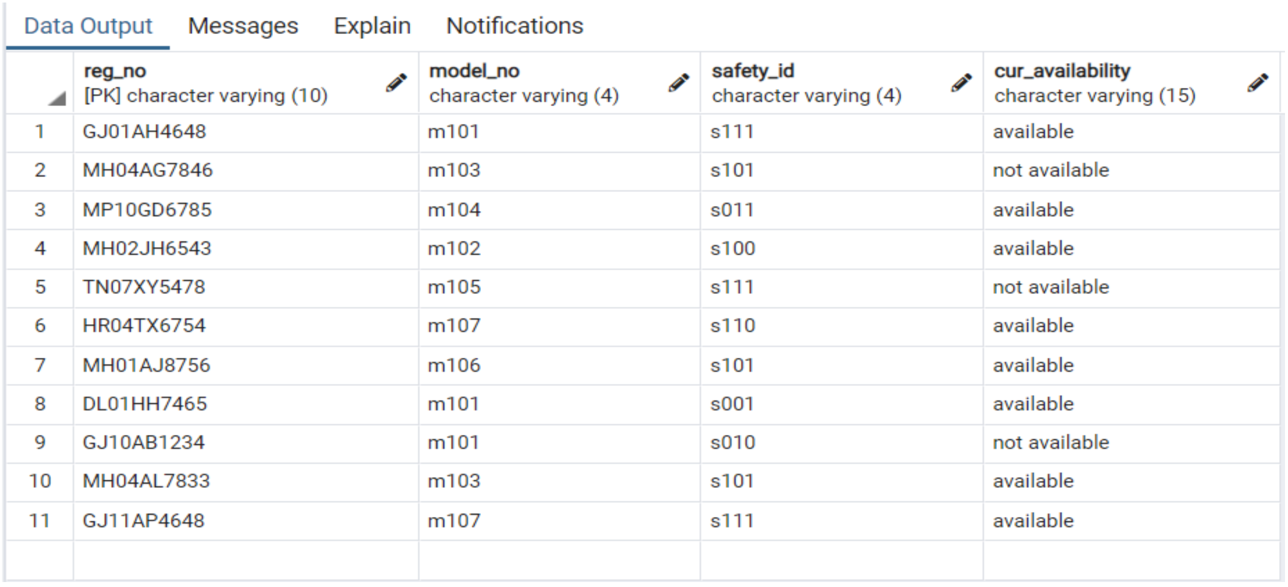


Data Output | Messages | Explain | Notifications

	cust_id [PK] character varying (4)	name character varying (30)	address character varying	phone text[]	license character varying (15)
1	c101	Harsh Sanghvi	Umiyaji Krupa, street n...	{784445...	GJ0312345678910
2	c156	Yash Patel	B101, Patel street, near...	{876543...	GJ1075876543210
3	c165	Jay Shah	55/d, Dhirendra Nath G...	{653165...	WB0185674326541
4	c201	Keval Desai	33, Dr Ambedkar Road,...	{765854...	MH0467854367285
5	c256	Rohit Sharma	22-5-227/a/g, Kothi, Hy...	{653423...	AP0164874532764
6	c109	Robin Patel	15,Chitrakulameast, M...	{738654...	TN0526547894567
7	c110	Vinay Mehta	440/a, Yallawa Smruti,...	{986543...	MH0163875467825
8	c102	Nikki Patel	Sai Niketan, Opp Rly St...	{764323...	MH0176352784563
9	c108	Sweta Tanna	934/911, Mittal Towers...	{654321...	KN0156372854628

5.2.3 Car

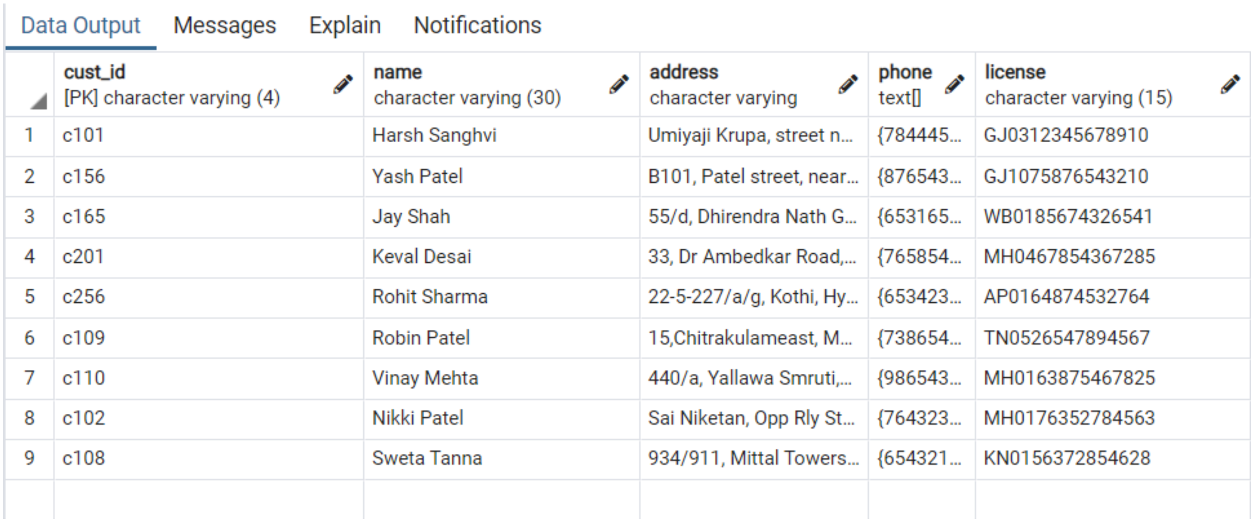
```
INSERT INTO car values

('GJ01AH4648','m101','s111','available'),

('MH04AG7846','m103','s101','not available'),

('MP10GD6785','m104','s011','available')

('MH02JH6543','m102','s100','available'),

('TN07XY5478','m105','s111','not available'),

('HR04TX6754','m107','s110','available'),

('MH01AJ8756','m106','s101','available'),

('DL01HH7465','m101','s001','available'),

('GJ10AB1234','m101','s010','not available')

('MH04AL7833','m103','s101','available'),

('GJ11AP4648','m107','s111','available');

INSERT 0 11
```



	reg_no [PK] character varying (10)	model_no character varying (4)	safety_id character varying (4)	cur_availability character varying (15)
1	GJ01AH4648	m101	s111	available
2	MH04AG7846	m103	s101	not available
3	MP10GD6785	m104	s011	available
4	MH02JH6543	m102	s100	available
5	TN07XY5478	m105	s111	not available
6	HR04TX6754	m107	s110	available
7	MH01AJ8756	m106	s101	available
8	DL01HH7465	m101	s001	available
9	GJ10AB1234	m101	s010	not available
10	MH04AL7833	m103	s101	available
11	GJ11AP4648	m107	s111	available

5.2.4 Car Category

```
INSERT INTO car_category values

('m101','Swift','5','7'),

('m102','XUV','7','9'),

('m103','Alto','5','6'),

('m104','SwiftDzire','5','8'),

('m105','Skoda','5','7'),

('m106','Innova','7','9'),

('m107','Cruze','5','10');

INSERT 0 7
```



	cust_id [PK] character varying (4)	name character varying (30)	address character varying	phone text[]	license character varying (15)
1	c101	Harsh Sanghvi	Umiyaji Krupa, street n...	{784445...	GJ0312345678910
2	c156	Yash Patel	B101, Patel street, near...	{876543...	GJ1075876543210
3	c165	Jay Shah	55/d, Dhirendra Nath G...	{653165...	WB0185674326541
4	c201	Keval Desai	33, Dr Ambedkar Road,...	{765854...	MH0467854367285
5	c256	Rohit Sharma	22-5-227/a/g, Kothi, Hy...	{653423...	AP0164874532764
6	c109	Robin Patel	15,Chitrakulameast, M...	{738654...	TN0526547894567
7	c110	Vinay Mehta	440/a, Yallawa Smruti,...	{986543...	MH0163875467825
8	c102	Nikki Patel	Sai Niketan, Opp Rly St...	{764323...	MH0176352784563
9	c108	Sweta Tanna	934/911, Mittal Towers...	{654321...	KN0156372854628

5.2.5 Safety Measures

INSERT INTO safety_measures values

('s000','N','N','N'),

('s001','N','N','Y'),

('s010','N','Y','N'),

('s011','N','Y','Y'),

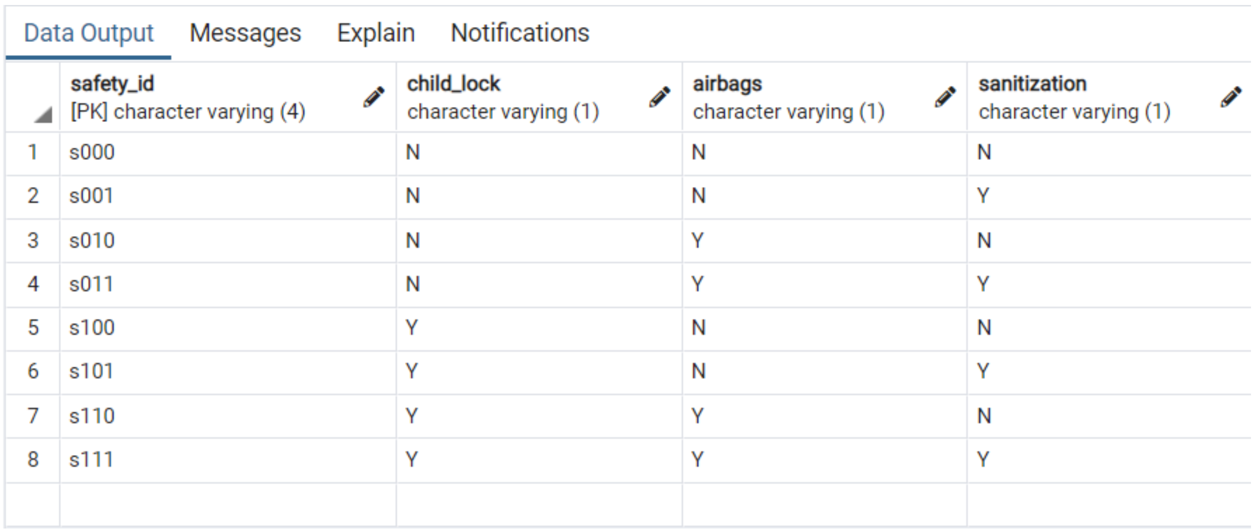
('s100','Y','N','N'),

('s101','Y','N','Y'),

('s110','Y','Y','N'),

('s111','Y','Y','Y');

INSERT 0 8

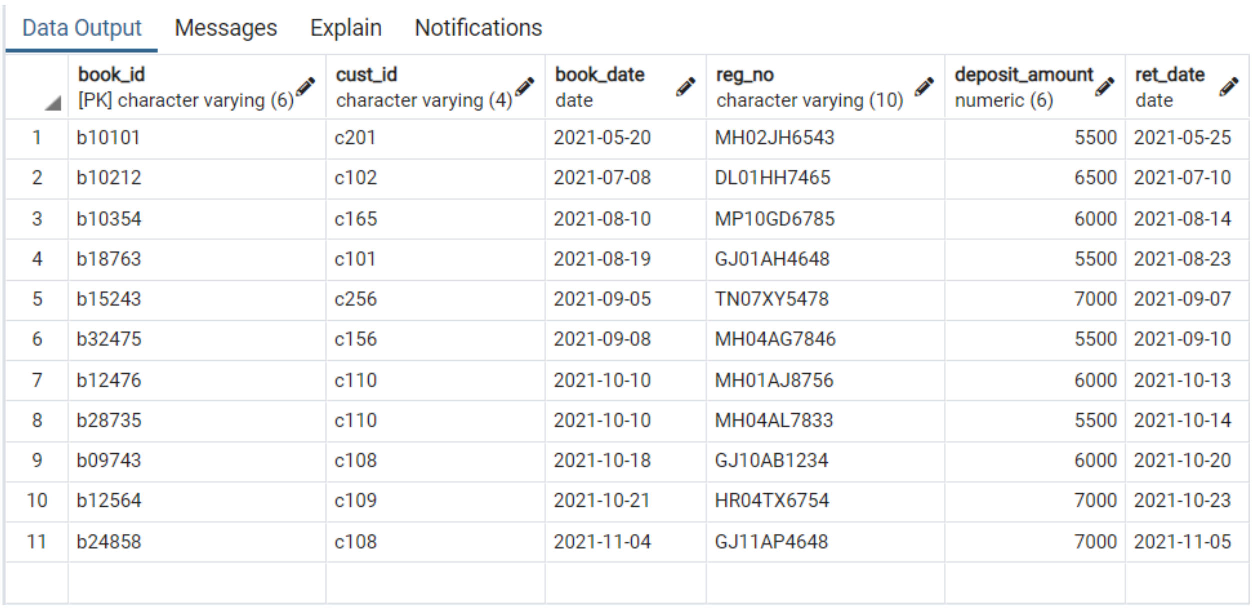


Data Output | Messages | Explain | Notifications

	safety_id [PK] character varying (4)	child_lock character varying (1)	airbags character varying (1)	sanitization character varying (1)
1	s000	N	N	N
2	s001	N	N	Y
3	s010	N	Y	N
4	s011	N	Y	Y
5	s100	Y	N	N
6	s101	Y	N	Y
7	s110	Y	Y	N
8	s111	Y	Y	Y

5.2.6 Booking

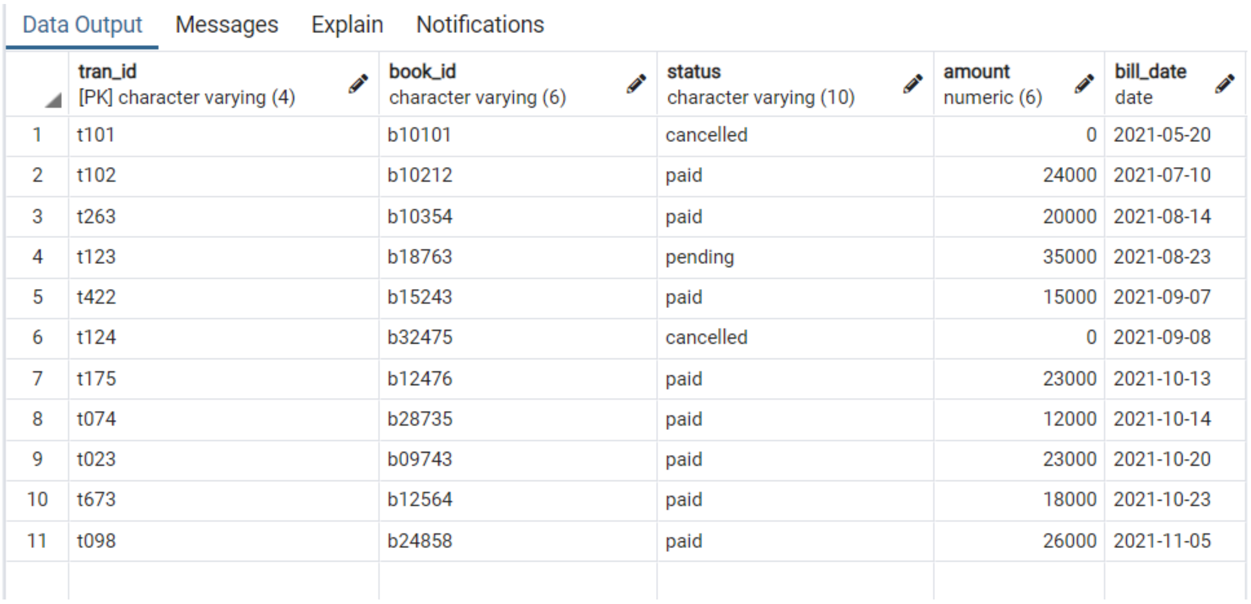
```
INSERT into booking values

('b10101','c201','20/05/2021','MH02JH6543','5500','25/05/2021'),

('b10212','c102','08/07/2021','DL01HH7465','6500','10/07/2021'),

('b10354','c165','10/08/2021','MP10GD6785','6000','14/08/2021'),

('b18763','c101','19/08/2021','GJ01AH4648','5500','23/08/2021'),

('b15243','c256','05/09/2021','TN07XY5478','7000','07/09/2021'),

('b32475','c156','08/09/2021','MH04AG7846','5500','10/09/2021'),

('b12476','c110','10/10/2021','MH01AJ8756','6000','13/10/2021'),

('b28735','c110','10/10/2021','MH04AL7833','5500','14/10/2021'),

('b09743','c108','18/10/2021','GJ10AB1234','6000','20/10/2021'),

('b12564','c109','21/10/2021','HR04TX6754','7000','23/10/2021'),

('b24858','c108','04/11/2021','GJ11AP4648','7000','05/11/2021');

INSERT 0 11
```



	book_id [PK] character varying (6)	cust_id character varying (4)	book_date date	reg_no character varying (10)	deposit_amount numeric (6)	ret_date date
1	b10101	c201	2021-05-20	MH02JH6543	5500	2021-05-25
2	b10212	c102	2021-07-08	DL01HH7465	6500	2021-07-10
3	b10354	c165	2021-08-10	MP10GD6785	6000	2021-08-14
4	b18763	c101	2021-08-19	GJ01AH4648	5500	2021-08-23
5	b15243	c256	2021-09-05	TN07XY5478	7000	2021-09-07
6	b32475	c156	2021-09-08	MH04AG7846	5500	2021-09-10
7	b12476	c110	2021-10-10	MH01AJ8756	6000	2021-10-13
8	b28735	c110	2021-10-10	MH04AL7833	5500	2021-10-14
9	b09743	c108	2021-10-18	GJ10AB1234	6000	2021-10-20
10	b12564	c109	2021-10-21	HR04TX6754	7000	2021-10-23
11	b24858	c108	2021-11-04	GJ11AP4648	7000	2021-11-05

5.2.7 Billing

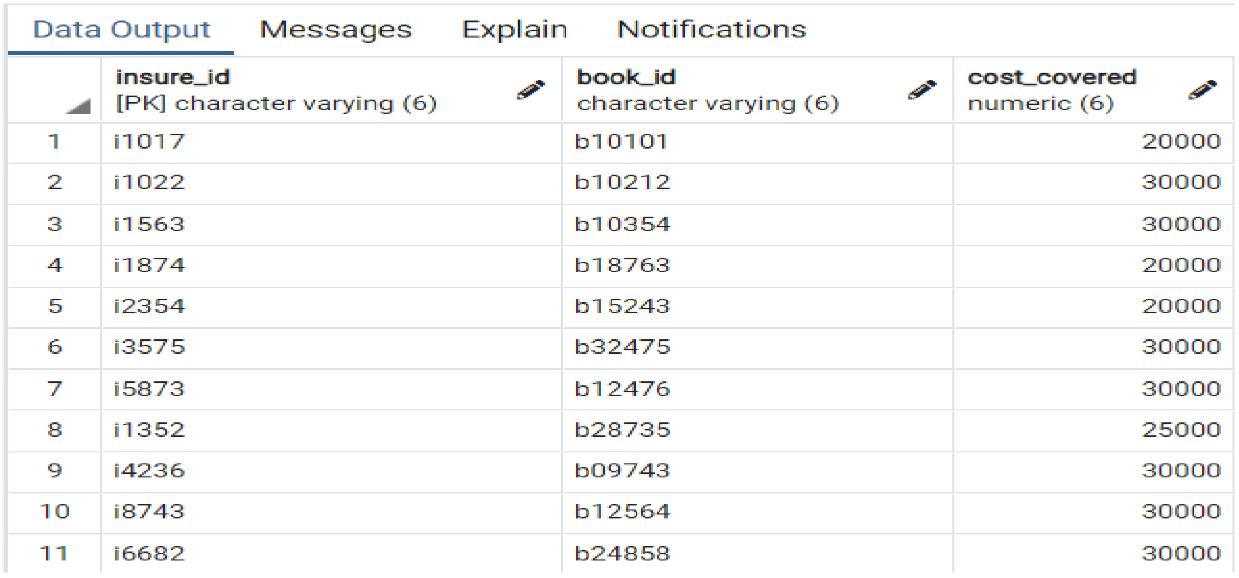
```
INSERT INTO billing values
('t101','b10101','cancelled','0','20/05/2021')

('t102','b10212','paid','24000','10/07/2021'),

('t263','b10354','paid','20000','14/08/2021'),

('t123','b18763','pending','35000','23/08/2021'),

('t422','b15243','paid','15000','07/09/2021'),

('t124','b32475','cancelled','0','08/09/2021'),

('t175','b12476','paid','23000','13/10/2021'),

('t074','b28735','paid','12000','14/10/2021'),

('t023','b09743','paid','23000','20/10/2021')

('t673','b12564','paid','18000','23/10/2021'),

('t098','b24858','paid','26000','05/11/2021');

INSERT 0 11
```



	tran_id [PK] character varying (4)	book_id character varying (6)	status character varying (10)	amount numeric (6)	bill_date date
1	t101	b10101	cancelled	0	2021-05-20
2	t102	b10212	paid	24000	2021-07-10
3	t263	b10354	paid	20000	2021-08-14
4	t123	b18763	pending	35000	2021-08-23
5	t422	b15243	paid	15000	2021-09-07
6	t124	b32475	cancelled	0	2021-09-08
7	t175	b12476	paid	23000	2021-10-13
8	t074	b28735	paid	12000	2021-10-14
9	t023	b09743	paid	23000	2021-10-20
10	t673	b12564	paid	18000	2021-10-23
11	t098	b24858	paid	26000	2021-11-05

5.2.8 Insurance

```
INSERT INTO insurance values

('i1017','b10101','20000'),

('i1022','b10212','30000'),

('i1563','b10354','30000')

,('i1874','b18763','20000'),

('i2354','b15243','20000'),

('i3575','b32475','30000'),

('i5873','b12476','30000'),

('i1352','b28735','25000'),

('i4236','b09743','30000'),

('i8743','b12564','30000'),

('i6682','b24858','30000');

INSERT 0 11
```



Data Output | Messages | Explain | Notifications

	insure_id [PK] character varying (6)	book_id character varying (6)	cost_covered numeric (6)
1	i1017	b10101	20000
2	i1022	b10212	30000
3	i1563	b10354	30000
4	i1874	b18763	20000
5	i2354	b15243	20000
6	i3575	b32475	30000
7	i5873	b12476	30000
8	i1352	b28735	25000
9	i4236	b09743	30000
10	i8743	b12564	30000
11	i6682	b24858	30000

5.2.9 Location

INSERT INTO location values

('l101','Veer Garage','Ring Road','Surat','Gujarat','395002'),

('l102','Manshukh stationery','Manek Chowk','Ahmedabad','Gujarat','380001')

('l301','Sports complex','Andheri','Mumbai','Maharashtra','400052'),

('l302','Hotel Western','North Main Road','Pune','Maharashtra','411016'),

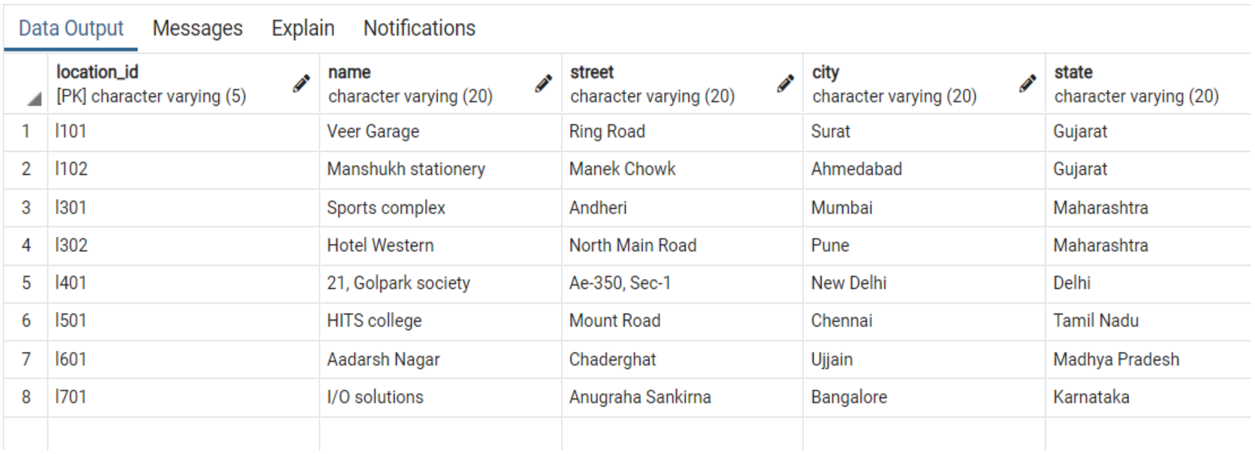
('l401','21, Golpark society','Ae-350, Sec-1','New Delhi','Delhi','700064'),

('l501','HITS college','Mount Road','Chennai','Tamil Nadu','600002'),

('l601','Aadarsh Nagar','Chaderghat','Ujjain','Madhya Pradesh','500024'),

('l701','I/O solutions','Anugraha Sankirna','Bangalore','Karnataka','560085');

INSERT 0 8

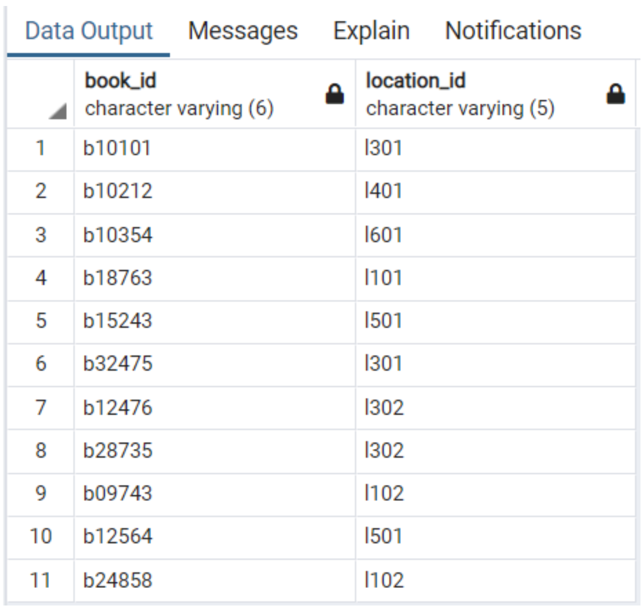


Data Output Messages Explain Notifications

	location_id [PK] character varying (5)	name character varying (20)	street character varying (20)	city character varying (20)	state character varying (20)
1	l101	Veer Garage	Ring Road	Surat	Gujarat
2	l102	Manshukh stationery	Manek Chowk	Ahmedabad	Gujarat
3	l301	Sports complex	Andheri	Mumbai	Maharashtra
4	l302	Hotel Western	North Main Road	Pune	Maharashtra
5	l401	21, Golpark society	Ae-350, Sec-1	New Delhi	Delhi
6	l501	HITS college	Mount Road	Chennai	Tamil Nadu
7	l601	Aadarsh Nagar	Chaderghat	Ujjain	Madhya Pradesh
8	l701	I/O solutions	Anugraha Sankirna	Bangalore	Karnataka

5.2.10 Pick-up

```
INSERT INTO pick_up values
('b10101','l301'),
('b10212','l401'),
('b10354','l601'),
('b18763','l101'),
('b15243','l501'),
('b32475','l301'),
('b12476','l302'),
('b28735','l302'),
('b09743','l102'),
('b12564','l501'),
('b24858','l102');
```



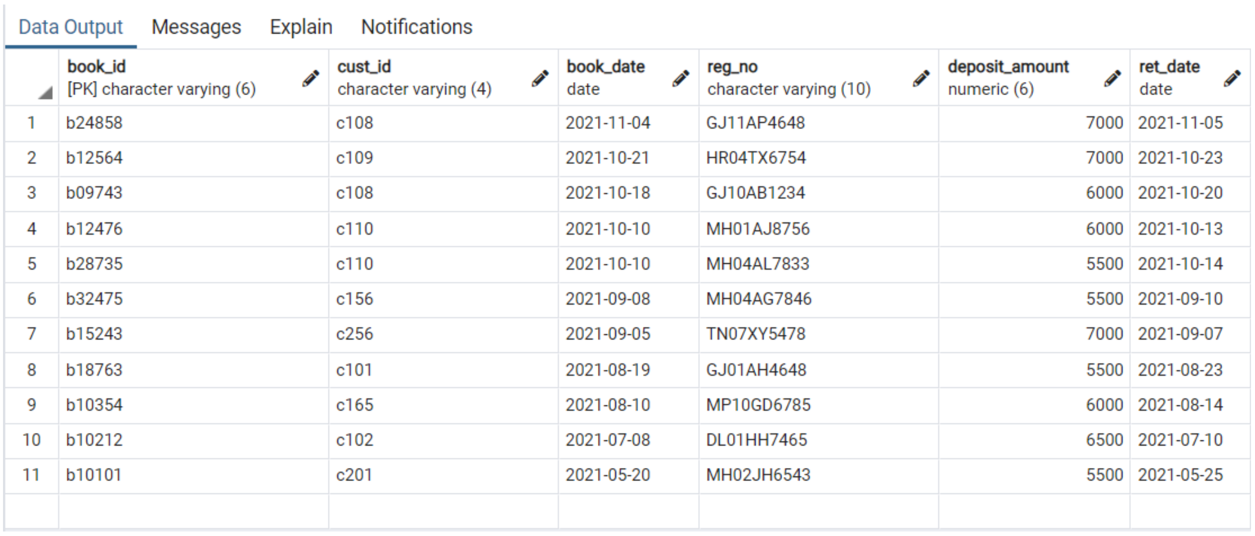
Data Output | Messages | Explain | Notifications

	book_id character varying (6)	location_id character varying (5)
1	b10101	l301
2	b10212	l401
3	b10354	l601
4	b18763	l101
5	b15243	l501
6	b32475	l301
7	b12476	l302
8	b28735	l302
9	b09743	l102
10	b12564	l501
11	b24858	l102

5.3 QUERIES

5.3.1 List all the information of bookings in descending order according to their booking dates.

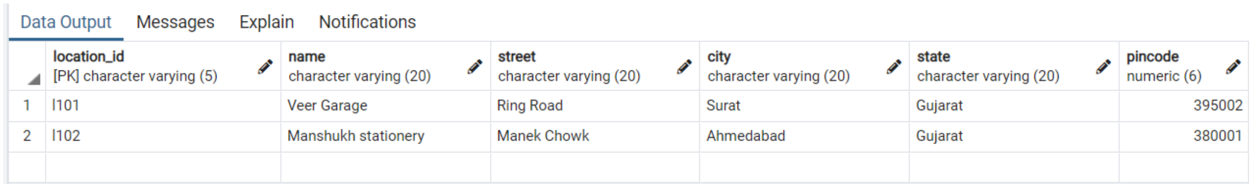
select * from booking ORDER BY book_date DESC;



	book_id [PK] character varying (6)	cust_id character varying (4)	book_date date	reg_no character varying (10)	deposit_amount numeric (6)	ret_date date
1	b24858	c108	2021-11-04	GJ11AP4648	7000	2021-11-05
2	b12564	c109	2021-10-21	HR04TX6754	7000	2021-10-23
3	b09743	c108	2021-10-18	GJ10AB1234	6000	2021-10-20
4	b12476	c110	2021-10-10	MH01AJ8756	6000	2021-10-13
5	b28735	c110	2021-10-10	MH04AL7833	5500	2021-10-14
6	b32475	c156	2021-09-08	MH04AG7846	5500	2021-09-10
7	b15243	c256	2021-09-05	TN07XY5478	7000	2021-09-07
8	b18763	c101	2021-08-19	GJ01AH4648	5500	2021-08-23
9	b10354	c165	2021-08-10	MP10GD6785	6000	2021-08-14
10	b10212	c102	2021-07-08	DL01HH7465	6500	2021-07-10
11	b10101	c201	2021-05-20	MH02JH6543	5500	2021-05-25

5.3.2 Display location information of pick up points existing in Gujarat.

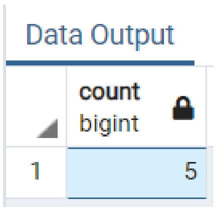
select * from location where state like 'Gujarat';



	location_id [PK] character varying (5)	name character varying (20)	street character varying (20)	city character varying (20)	state character varying (20)	pincode numeric (6)
1	l101	Veer Garage	Ring Road	Surat	Gujarat	395002
2	l102	Manshukh stationery	Manek Chowk	Ahmedabad	Gujarat	380001

5.3.3 Count no. of bills where amount is greater than 20,000.

select count(*) from billing where amount>20000;

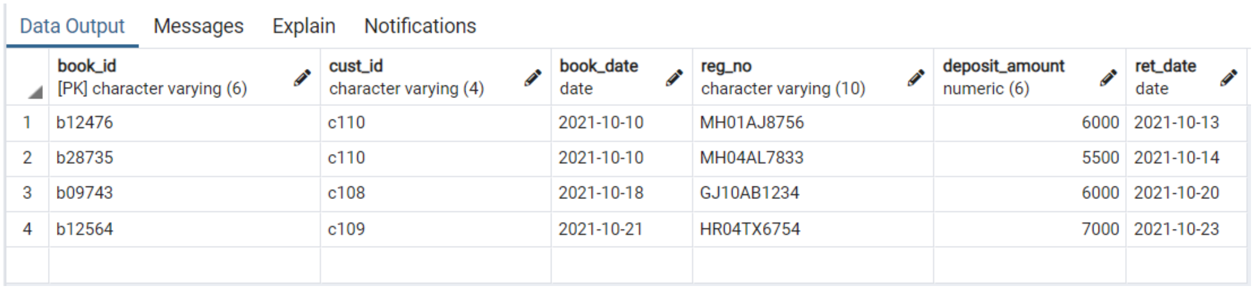


Data Output

	count bigint
1	5

5.3.4 Display booking information of cars booked in the month of October.

select * from booking where book_date BETWEEN '1/10/2021' AND '31/10/2021';

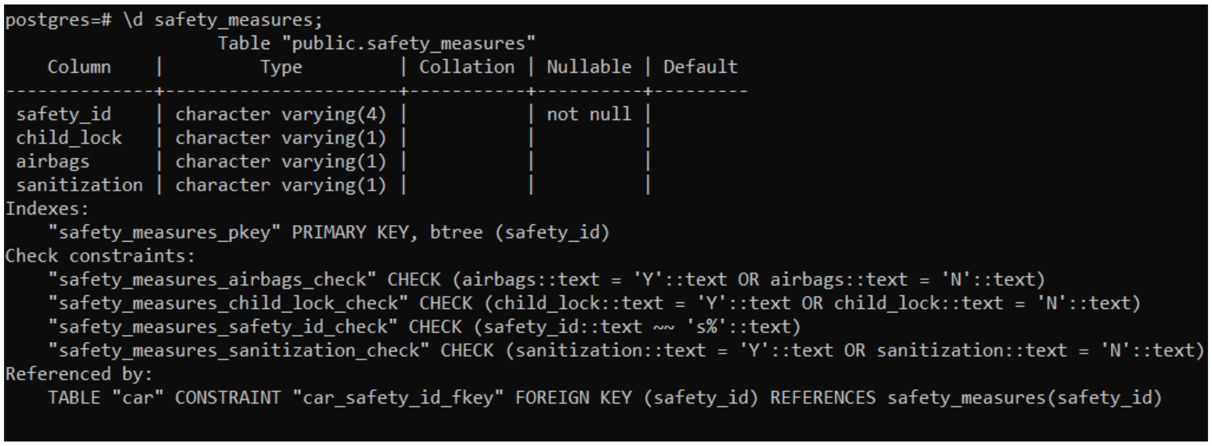


Data Output Messages Explain Notifications

	book_id [PK] character varying (6)	cust_id character varying (4)	book_date date	reg_no character varying (10)	deposit_amount numeric (6)	ret_date date
1	b12476	c110	2021-10-10	MH01AJ8756	6000	2021-10-13
2	b28735	c110	2021-10-10	MH04AL7833	5500	2021-10-14
3	b09743	c108	2021-10-18	GJ10AB1234	6000	2021-10-20
4	b12564	c109	2021-10-21	HR04TX6754	7000	2021-10-23

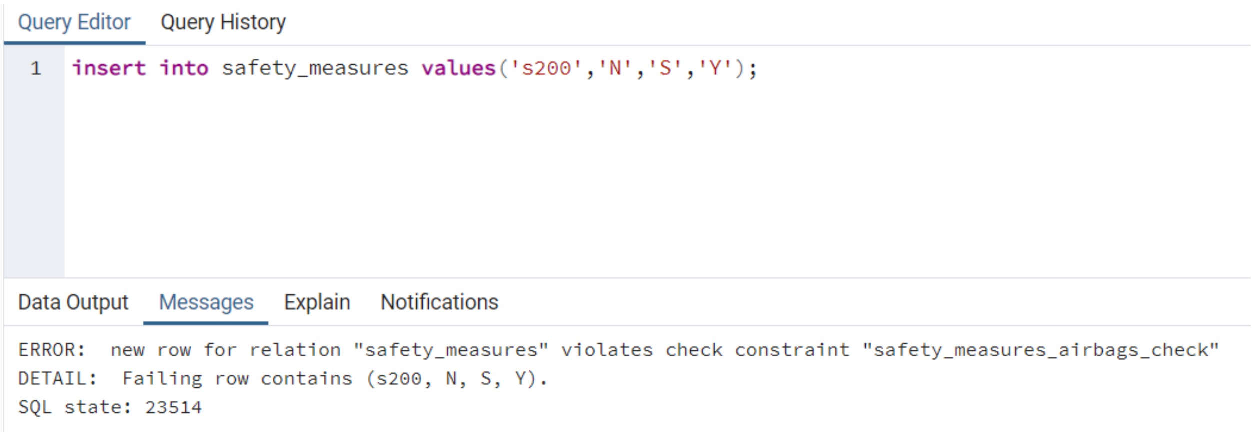
5.3.5 Try inserting any value other than ‘Y’ and ‘N’ in safety measures having check constraints:

postgres=# CREATE TABLE safety_measures(
postgres(# safety_id varchar(4) primary key CHECK (safety_id like 's%'),
postgres(# child_lock varchar(1) CHECK (child_lock='Y' or child_lock='N'),
postgres(# airbags varchar(1) CHECK (airbags='Y' or airbags='N'),
postgres(# sanitization varchar(1) CHECK (sanitization='Y' or sanitization='N'));



```
postgres=# \d safety_measures;
                    Table "public.safety_measures"
    Column    |         Type         | Collation | Nullable | Default
--------------+----------------------+-----------+----------+---------
 safety_id    | character varying(4) |           | not null |
 child_lock   | character varying(1) |           |          |
 airbags      | character varying(1) |           |          |
 sanitization | character varying(1) |           |          |
Indexes:
    "safety_measures_pkey" PRIMARY KEY, btree (safety_id)
Check constraints:
    "safety_measures_airbags_check" CHECK (airbags::text = 'Y'::text OR airbags::text = 'N'::text)
    "safety_measures_child_lock_check" CHECK (child_lock::text = 'Y'::text OR child_lock::text = 'N'::text)
    "safety_measures_safety_id_check" CHECK (safety_id::text ~~ 's%'::text)
    "safety_measures_sanitization_check" CHECK (sanitization::text = 'Y'::text OR sanitization::text = 'N'::text)
Referenced by:
    TABLE "car" CONSTRAINT "car_safety_id_fkey" FOREIGN KEY (safety_id) REFERENCES safety_measures(safety_id)
```

Query:
insert into safety_measures values('s200','N','S','Y');

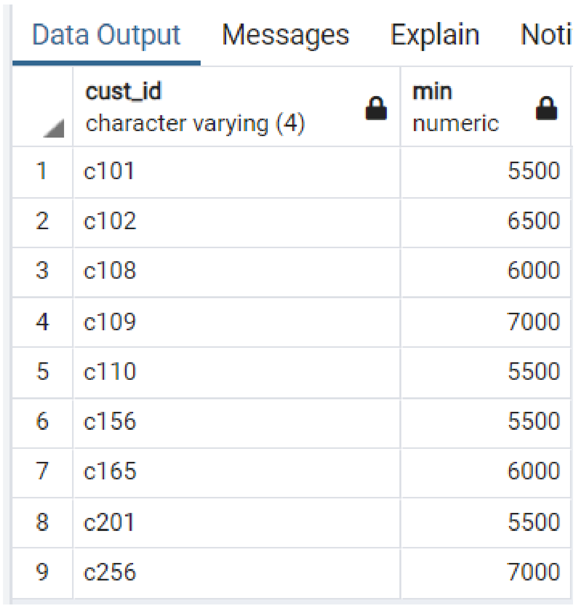


```
insert into safety_measures values('s200','N','S','Y');
```

```
ERROR:  new row for relation "safety_measures" violates check constraint "safety_measures_airbags_check"
DETAIL:  Failing row contains (s200, N, S, Y).
SQL state: 23514
```

5.3.6 Display minimum deposit amount of each customer id in booking table.

select cust_id,min(deposit_amount) from booking group by cust_id order by cust_id;

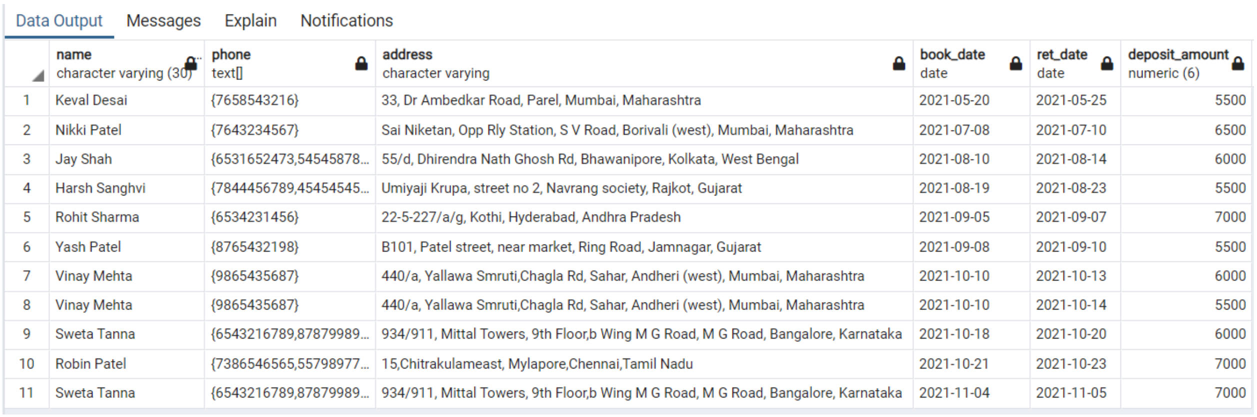


	cust_id character varying (4)	min numeric
1	c101	5500
2	c102	6500
3	c108	6000
4	c109	7000
5	c110	5500
6	c156	5500
7	c165	6000
8	c201	5500
9	c256	7000

Join Queries:

5.3.7 Display customer name, address, phone number, booking date, return date and deposit amount using joins.

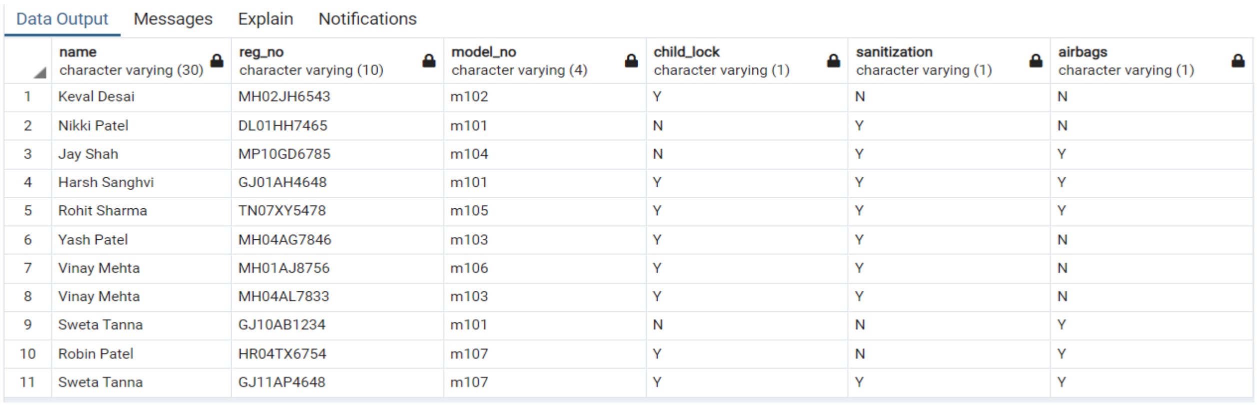
select customer.name,customer.phone,customer.address,booking.book_date,booking.ret_date,booking.deposit_amount from customer LEFT JOIN booking on customer.cust_id=booking.cust_id;



	name character varying (30)	phone text[]	address character varying	book_date date	ret_date date	deposit_amount numeric (6)
1	Keval Desai	{7658543216}	33, Dr Ambedkar Road, Parel, Mumbai, Maharashtra	2021-05-20	2021-05-25	5500
2	Nikki Patel	{7643234567}	Sai Niketan, Opp Rly Station, S V Road, Borivali (west), Mumbai, Maharashtra	2021-07-08	2021-07-10	6500
3	Jay Shah	{6531652473,54545878…	55/d, Dhirendra Nath Ghosh Rd, Bhawanipore, Kolkata, West Bengal	2021-08-10	2021-08-14	6000
4	Harsh Sanghvi	{7844456789,45454545…	Umiyaji Krupa, street no 2, Navrang society, Rajkot, Gujarat	2021-08-19	2021-08-23	5500
5	Rohit Sharma	{6534231456}	22-5-227/a/g, Kothi, Hyderabad, Andhra Pradesh	2021-09-05	2021-09-07	7000
6	Yash Patel	{8765432198}	B101, Patel street, near market, Ring Road, Jamnagar, Gujarat	2021-09-08	2021-09-10	5500
7	Vinay Mehta	{9865435687}	440/a, Yallawa Smruti,Chagla Rd, Sahar, Andheri (west), Mumbai, Maharashtra	2021-10-10	2021-10-13	6000
8	Vinay Mehta	{9865435687}	440/a, Yallawa Smruti,Chagla Rd, Sahar, Andheri (west), Mumbai, Maharashtra	2021-10-10	2021-10-14	5500
9	Sweta Tanna	{6543216789,87879989…	934/911, Mittal Towers, 9th Floor,b Wing M G Road, M G Road, Bangalore, Karnataka	2021-10-18	2021-10-20	6000
10	Robin Patel	{7386546565,55798977…	15,Chitrakulameast, Mylapore,Chennai,Tamil Nadu	2021-10-21	2021-10-23	7000
11	Sweta Tanna	{6543216789,87879989…	934/911, Mittal Towers, 9th Floor,b Wing M G Road, M G Road, Bangalore, Karnataka	2021-11-04	2021-11-05	7000

5.3.8 Display name of customers and reg no, model no and safety features of a cars booked by them.

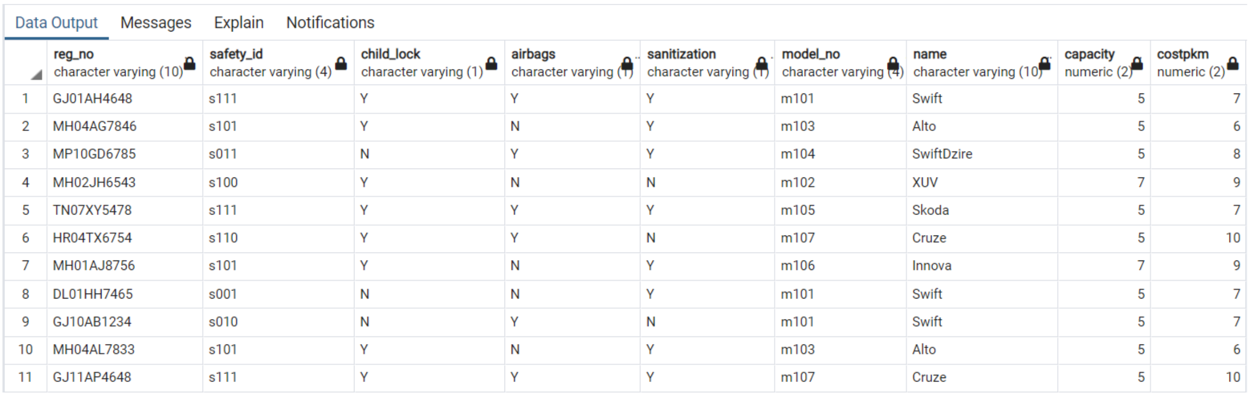
select customer.name,car.reg_no,car.model_no,safety_measures.child_lock,safety_measures.sanitization,safety_measures.airbags from customer inner join booking on customer.cust_id=booking.cust_id inner join car on car.reg_no=booking.reg_no inner join safety_measures on car.safety_id=safety_measures.safety_id;



	name character varying (30)	reg_no character varying (10)	model_no character varying (4)	child_lock character varying (1)	sanitization character varying (1)	airbags character varying (1)
1	Keval Desai	MH02JH6543	m102	Y	N	N
2	Nikki Patel	DL01HH7465	m101	N	Y	N
3	Jay Shah	MP10GD6785	m104	N	Y	Y
4	Harsh Sanghvi	GJ01AH4648	m101	Y	Y	Y
5	Rohit Sharma	TN07XY5478	m105	Y	Y	Y
6	Yash Patel	MH04AG7846	m103	Y	Y	N
7	Vinay Mehta	MH01AJ8756	m106	Y	Y	N
8	Vinay Mehta	MH04AL7833	m103	Y	Y	N
9	Sweta Tanna	GJ10AB1234	m101	N	N	Y
10	Robin Patel	HR04TX6754	m107	Y	N	Y
11	Sweta Tanna	GJ11AP4648	m107	Y	Y	Y

5.3.9 Display car reg no, safety measures and category using joins.

select car.reg_no, safety_measures.*,car_category.* from car inner join safety_measures on car.safety_id=safety_measures.safety_id inner join car_category on car.model_no=car_category.model_no;

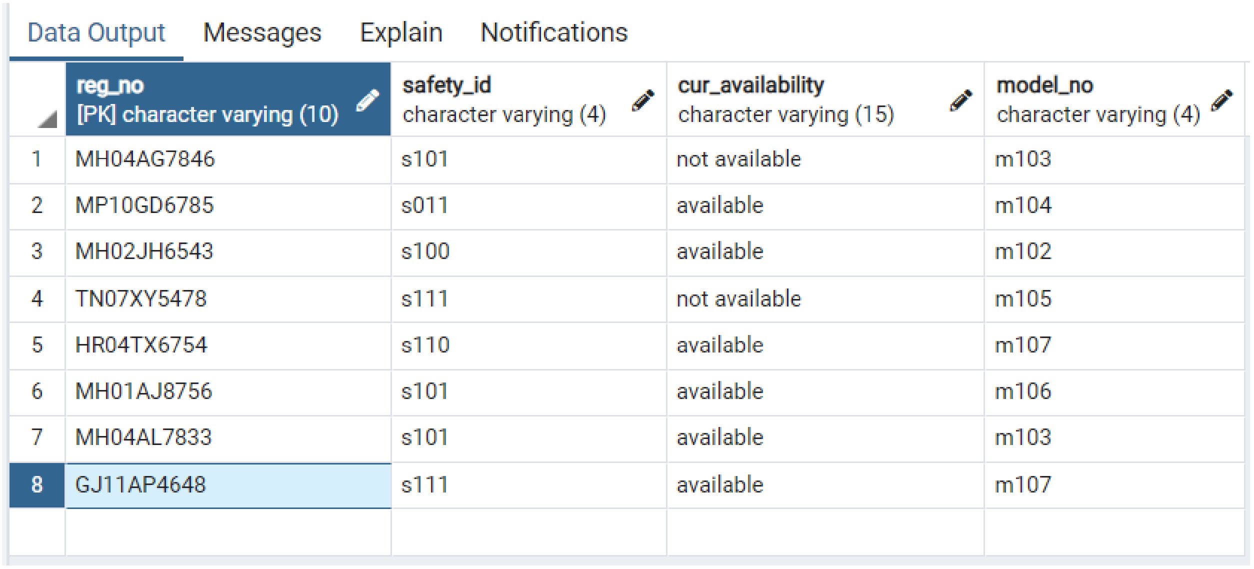


Data Output | Messages | Explain | Notifications

	reg_no character varying (10)	safety_id character varying (4)	child_lock character varying (1)	airbags character varying (1)	sanitization character varying (1)	model_no character varying (4)	name character varying (10)	capacity numeric (2)	costpkm numeric (2)
1	GJ01AH4648	s111	Y	Y	Y	m101	Swift	5	7
2	MH04AG7846	s101	Y	N	Y	m103	Alto	5	6
3	MP10GD6785	s011	N	Y	Y	m104	SwiftDzire	5	8
4	MH02JH6543	s100	Y	N	N	m102	XUV	7	9
5	TN07XY5478	s111	Y	Y	Y	m105	Skoda	5	7
6	HR04TX6754	s110	Y	Y	N	m107	Cruze	5	10
7	MH01AJ8756	s101	Y	N	Y	m106	Innova	7	9
8	DL01HH7465	s001	N	N	Y	m101	Swift	5	7
9	GJ10AB1234	s010	N	Y	N	m101	Swift	5	7
10	MH04AL7833	s101	Y	N	Y	m103	Alto	5	6
11	GJ11AP4648	s111	Y	Y	Y	m107	Cruze	5	10

5.3.10 Display car details of cars whose model no is not m101.

select c1.reg_no,c1.safety_id,c1.cur_availability,c1.model_no from car c1 INNER JOIN car c2 on c1.model_no NOT IN (select c2.model_no from car c2 where c2.model_no='m101') AND c1.reg_no=c2.reg_no;

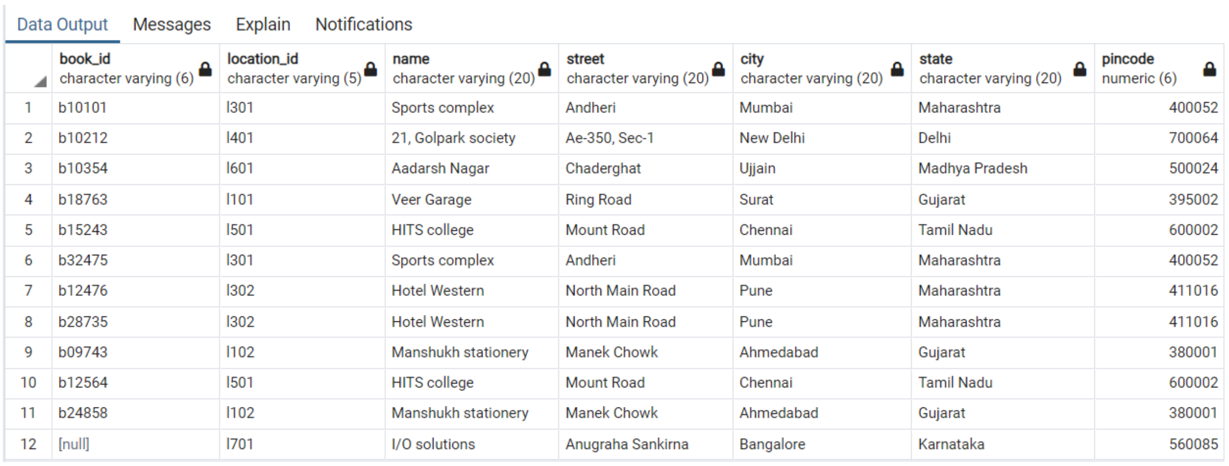


Data Output | Messages | Explain | Notifications

	reg_no [PK] character varying (10)	safety_id character varying (4)	cur_availability character varying (15)	model_no character varying (4)
1	MH04AG7846	s101	not available	m103
2	MP10GD6785	s011	available	m104
3	MH02JH6543	s100	available	m102
4	TN07XY5478	s111	not available	m105
5	HR04TX6754	s110	available	m107
6	MH01AJ8756	s101	available	m106
7	MH04AL7833	s101	available	m103
8	GJ11AP4648	s111	available	m107

5.3.11 Display all location information along with their corresponding booking id's.

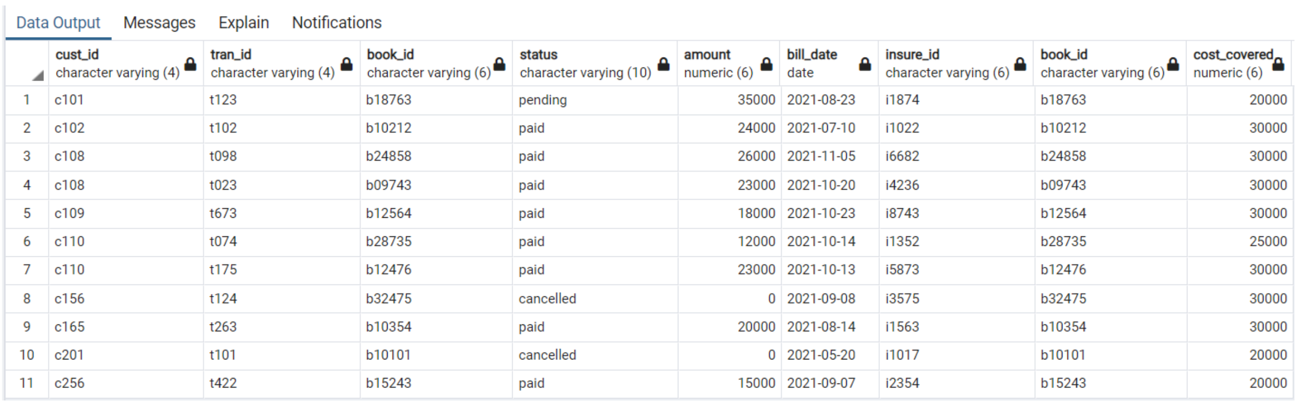
select booking.book_id,location.* from pick_up LEFT JOIN booking on booking.book_id=pick_up.book_id RIGHT JOIN location on pick_up.location_id=location.location_id;



	book_id character varying (6)	location_id character varying (5)	name character varying (20)	street character varying (20)	city character varying (20)	state character varying (20)	pincode numeric (6)
1	b10101	l301	Sports complex	Andheri	Mumbai	Maharashtra	400052
2	b10212	l401	21, Golpark society	Ae-350, Sec-1	New Delhi	Delhi	700064
3	b10354	l601	Aadarsh Nagar	Chaderghat	Ujjain	Madhya Pradesh	500024
4	b18763	l101	Veer Garage	Ring Road	Surat	Gujarat	395002
5	b15243	l501	HITS college	Mount Road	Chennai	Tamil Nadu	600002
6	b32475	l301	Sports complex	Andheri	Mumbai	Maharashtra	400052
7	b12476	l302	Hotel Western	North Main Road	Pune	Maharashtra	411016
8	b28735	l302	Hotel Western	North Main Road	Pune	Maharashtra	411016
9	b09743	l102	Manshukh stationery	Manek Chowk	Ahmedabad	Gujarat	380001
10	b12564	l501	HITS college	Mount Road	Chennai	Tamil Nadu	600002
11	b24858	l102	Manshukh stationery	Manek Chowk	Ahmedabad	Gujarat	380001
12	[null]	l701	I/O solutions	Anugraha Sankirna	Bangalore	Karnataka	560085

5.3.12 Display customer id and billing information and insurance information corresponding to their customer id.

select customer.cust_id,billing.*,insurance.* from customer INNER JOIN booking on customer.cust_id=booking.cust_id INNER JOIN billing on booking.book_id=billing.book_id INNER JOIN insurance on booking.book_id=insurance.book_id ORDER BY customer.cust_id;

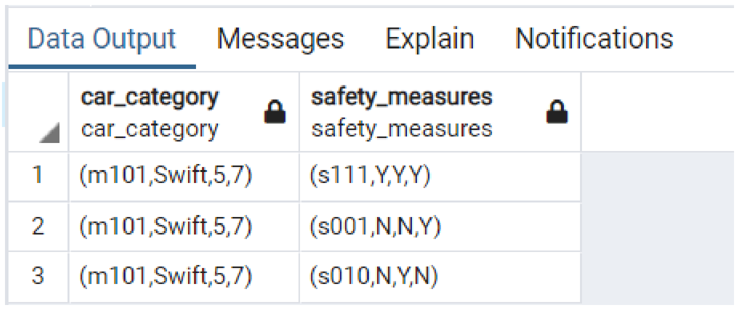


	cust_id character varying (4)	tran_id character varying (4)	book_id character varying (6)	status character varying (10)	amount numeric (6)	bill_date date	insure_id character varying (6)	book_id character varying (6)	cost_covered numeric (6)
1	c101	t123	b18763	pending	35000	2021-08-23	i1874	b18763	20000
2	c102	t102	b10212	paid	24000	2021-07-10	i1022	b10212	30000
3	c108	t098	b24858	paid	26000	2021-11-05	i6682	b24858	30000
4	c108	t023	b09743	paid	23000	2021-10-20	i4236	b09743	30000
5	c109	t673	b12564	paid	18000	2021-10-23	i8743	b12564	30000
6	c110	t074	b28735	paid	12000	2021-10-14	i1352	b28735	25000
7	c110	t175	b12476	paid	23000	2021-10-13	i5873	b12476	30000
8	c156	t124	b32475	cancelled	0	2021-09-08	i3575	b32475	30000
9	c165	t263	b10354	paid	20000	2021-08-14	i1563	b10354	30000
10	c201	t101	b10101	cancelled	0	2021-05-20	i1017	b10101	20000
11	c256	t422	b15243	paid	15000	2021-09-07	i2354	b15243	20000

Sub Queries:

5.3.13 Display car category and safety measures of cars where model numbers is m101.

SELECT car_category.,safety_measures. from car INNER JOIN safety_measures ON car.safety_id=safety_measures.safety_id INNER JOIN car_category ON car_category.model_no=car.model_no where (car.model_no like 'm101');

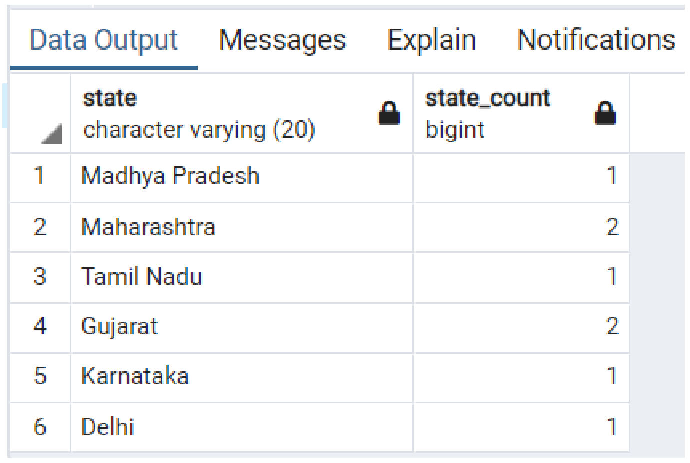


Data Output | Messages | Explain | Notifications

	car_category car_category	safety_measures safety_measures
1	(m101,Swift,5,7)	(s111,Y,Y,Y)
2	(m101,Swift,5,7)	(s001,N,N,Y)
3	(m101,Swift,5,7)	(s010,N,Y,N)

5.3.14 Display Count no. of pick-up points in a particular state.

SELECT l.state,count(l.state) as state_count from location l right outer join location k on l.pincode=k.pincode group by l.state;

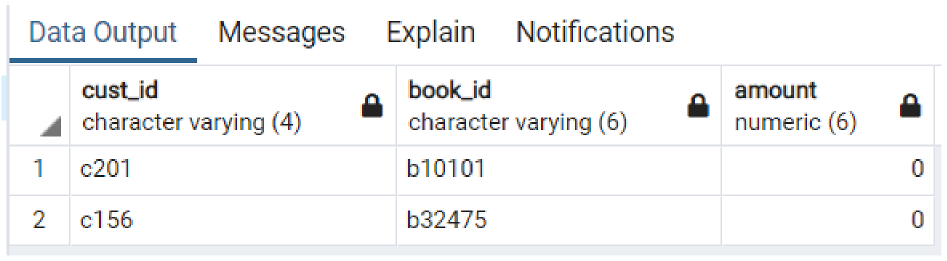


Data Output | Messages | Explain | Notifications

	state character varying (20)	state_count bigint
1	Madhya Pradesh	1
2	Maharashtra	2
3	Tamil Nadu	1
4	Gujarat	2
5	Karnataka	1
6	Delhi	1

5.3.15 Display booking id, customer id and billing amount of customers whose billing amount is minimum.

```
select booking.cust_id, booking.book_id,billing.amount from booking inner join
billing on booking.book_id=billing.book_id
where billing.amount=(select min(amount) from billing);
```

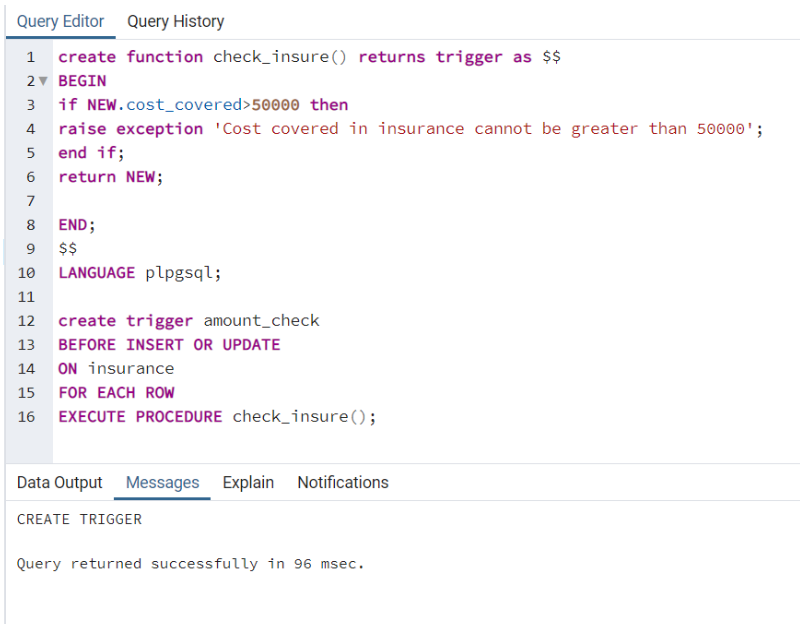


	cust_id character varying (4)	book_id character varying (6)	amount numeric (6)
1	c201	b10101	0
2	c156	b32475	0

5.4 TRIGGERS

5.4.1 Display Error message when customer tries to enter insurance amount more than 50,000.

```
create function check_insure() returns trigger as $$
BEGIN
if NEW.cost_covered>50000 then
raise exception 'Cost covered in insurance cannot be greater than 50000';
end if;
return NEW;

END;
$$
LANGUAGE plpgsql;

create trigger amount_check
BEFORE INSERT OR UPDATE
ON insurance
FOR EACH ROW
EXECUTE PROCEDURE check_insure();
```

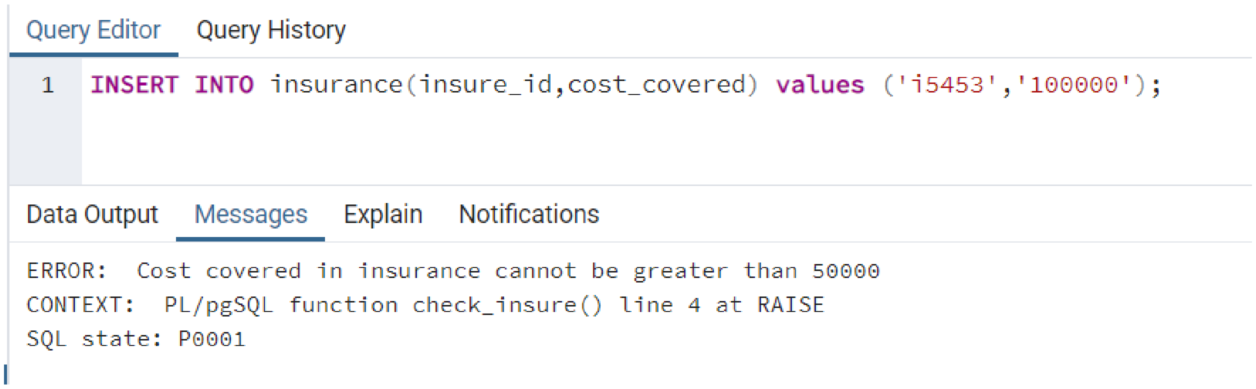


```
create function check_insure() returns trigger as $$
BEGIN
if NEW.cost_covered>50000 then
raise exception 'Cost covered in insurance cannot be greater than 50000';
end if;
return NEW;

END;
$$
LANGUAGE plpgsql;

create trigger amount_check
BEFORE INSERT OR UPDATE
ON insurance
FOR EACH ROW
EXECUTE PROCEDURE check_insure();
```

```
CREATE TRIGGER

Query returned successfully in 96 msec.
```

Query:

INSERT INTO insurance(insure_id,cost_covered) values ('i5453','100000');



```
INSERT INTO insurance(insure_id,cost_covered) values ('i5453','100000');
```

```
ERROR:  Cost covered in insurance cannot be greater than 50000
CONTEXT:  PL/pgSQL function check_insure() line 4 at RAISE
SQL state: P0001
```

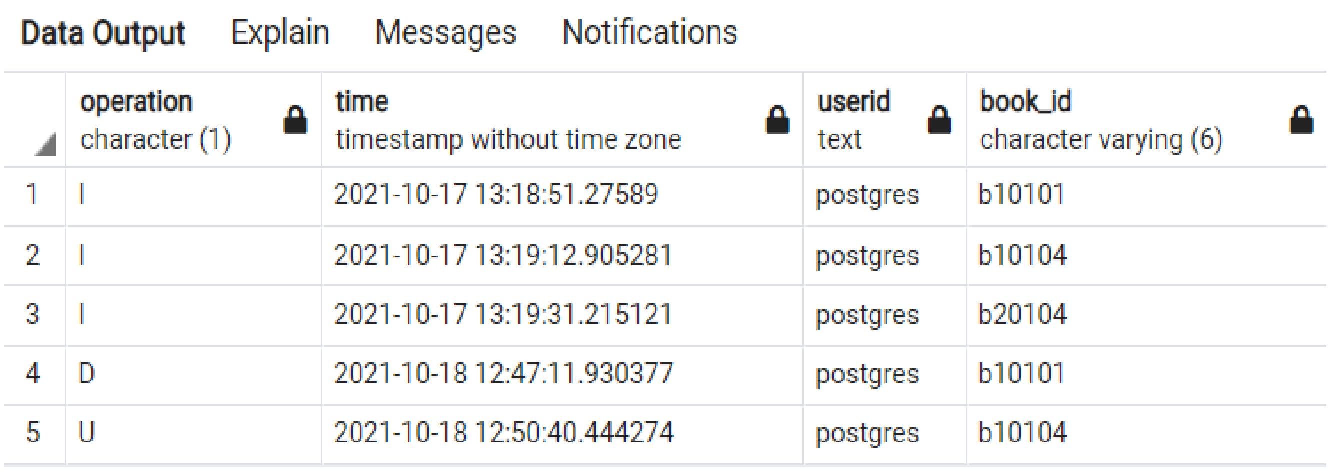
5.4.2Create an audit which keeps a track of records inserted, updated and deleted in booking table.

```
create table book_audit(
operation character(1) not null,
time timestamp not null,
userid text not null,
book_id varchar(6)
);

CREATE OR REPLACE FUNCTION do_book_audit()
RETURNS TRIGGER AS $$
BEGIN
IF (tg_op='DELETE') THEN
INSERT INTO book_audit SELECT 'D',
now(),user,OLD.book_id;
RETURN OLD;
ELSEIF(tg_op='UPDATE') THEN
INSERT INTO book_audit SELECT
'U',now(),user,NEW.book_id;
RETURN NEW;
ELSEIF(tg_op='INSERT') THEN
INSERT INTO book_audit SELECT
'I',now(),user,NEW.book_id;
RETURN NEW;
END IF;
RETURN NULL;
END;
$$
LANGUAGE plpgsql;

CREATE TRIGGER boo_audit
AFTER INSERT or UPDATE or DELETE ON booking
FOR EACH ROW EXECUTE PROCEDURE do_book_audit()
```

Queries:

DELETE FROM booking WHERE book_id like 'b10101';

UPDATE booking SET deposit_amount = '7000' WHERE book_id = 'b10104';



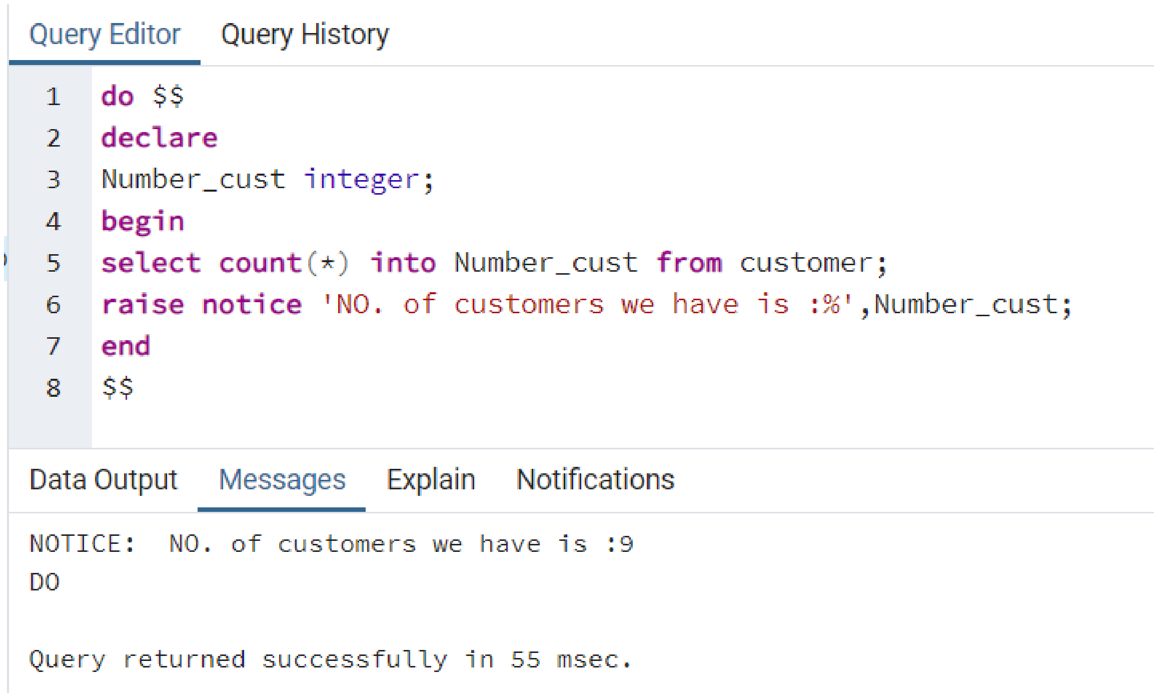
Data Output Explain Messages Notifications

	operation character (1)	time timestamp without time zone	userid text	book_id character varying (6)
1	I	2021-10-17 13:18:51.27589	postgres	b10101
2	I	2021-10-17 13:19:12.905281	postgres	b10104
3	I	2021-10-17 13:19:31.215121	postgres	b20104
4	D	2021-10-18 12:47:11.930377	postgres	b10101
5	U	2021-10-18 12:50:40.444274	postgres	b10104

5.5 PLSQL BLOCKS:

5.5.1 Display no. of customers in customer table.

```
do $$
declare
Number_cust integer;
begin
select count(*) into Number_cust from customer;
raise notice 'NO. of customers we have is :%',Number_cust;
end
$$
```

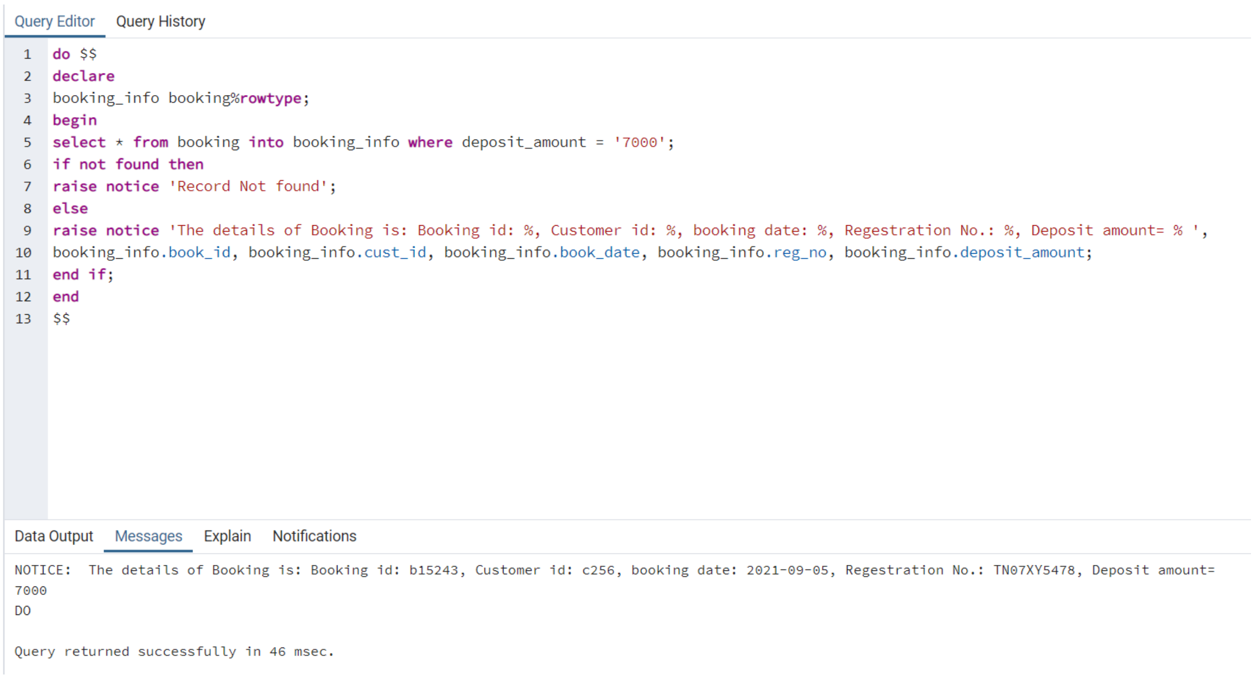


```
do $$
declare
Number_cust integer;
begin
select count(*) into Number_cust from customer;
raise notice 'NO. of customers we have is :%',Number_cust;
end
$$
```

```
NOTICE:  NO. of customers we have is :9
DO

Query returned successfully in 55 msec.
```

5.5.2 Display booking details where deposit amount is 7000 rs.

do $$
declare
booking_info booking%rowtype;
begin
select * from booking into booking_info where deposit_amount = '7000';
if not found then
raise notice 'Record Not found';
else
raise notice 'The details of Booking is: Booking id: %, Customer id: %, booking date: %, Regestration No.: %, Deposit amount= % ',
booking_info.book_id, booking_info.cust_id, booking_info.book_date,
booking_info.reg_no, booking_info.deposit_amount;
end if;
end
$$



```
do $$
declare
booking_info booking%rowtype;
begin
select * from booking into booking_info where deposit_amount = '7000';
if not found then
raise notice 'Record Not found';
else
raise notice 'The details of Booking is: Booking id: %, Customer id: %, booking date: %, Regestration No.: %, Deposit amount= % ',
booking_info.book_id, booking_info.cust_id, booking_info.book_date, booking_info.reg_no, booking_info.deposit_amount;
end if;
end
$$
```

```
NOTICE:  The details of Booking is: Booking id: b15243, Customer id: c256, booking date: 2021-09-05, Regestration No.: TN07XY5478, Deposit amount=
7000
DO

Query returned successfully in 46 msec.
```

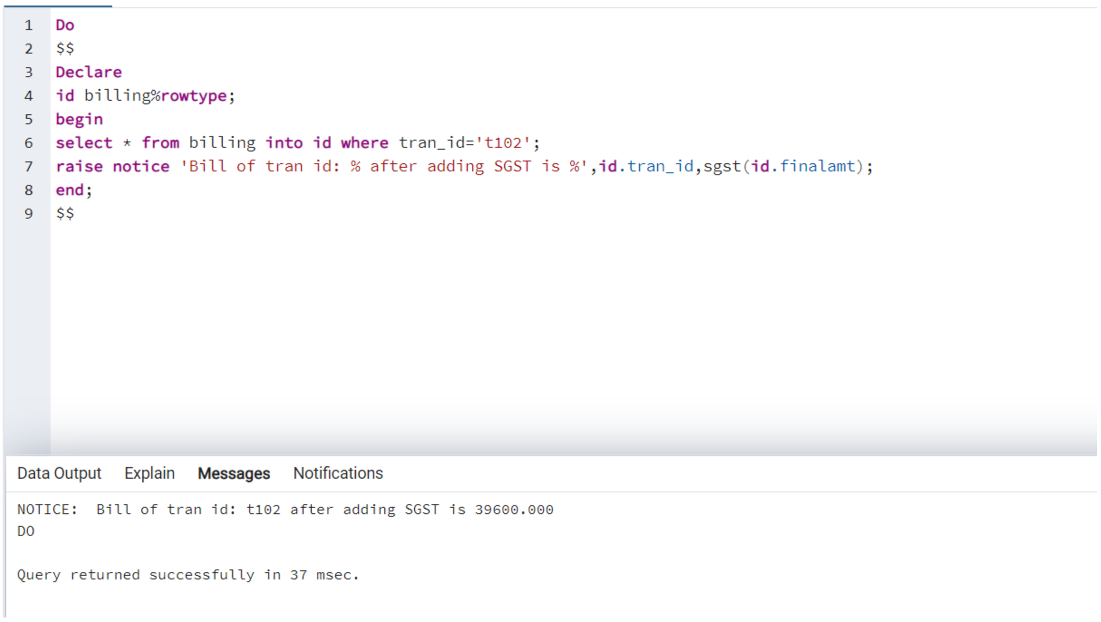
5.6 FUNCTION

5.6.1 Create a function to find the final billing value in billing table by adding 10 percent SGST to final amount.

```
create function sgst(finalamt numeric)
returns numeric
language plpgsql
as

$$
declare
total numeric;
begin
total = finalamt*0.10+finalamt;
return total;
end;
$$;
```

PL SQL block:

```
Do
$$
Declare
id billing%rowtype;
begin
select * from billing into id where tran_id='t102';
raise notice 'Bill of tran id: % after adding SGST is %',id.tran_id,sgst(id.finalamt);
end;
$$
```



```
Do
$$
Declare
id billing%rowtype;
begin
select * from billing into id where tran_id='t102';
raise notice 'Bill of tran id: % after adding SGST is %',id.tran_id,sgst(id.finalamt);
end;
$$
```

Data Output Explain **Messages** Notifications

```
NOTICE:  Bill of tran id: t102 after adding SGST is 39600.000
DO

Query returned successfully in 37 msec.
```

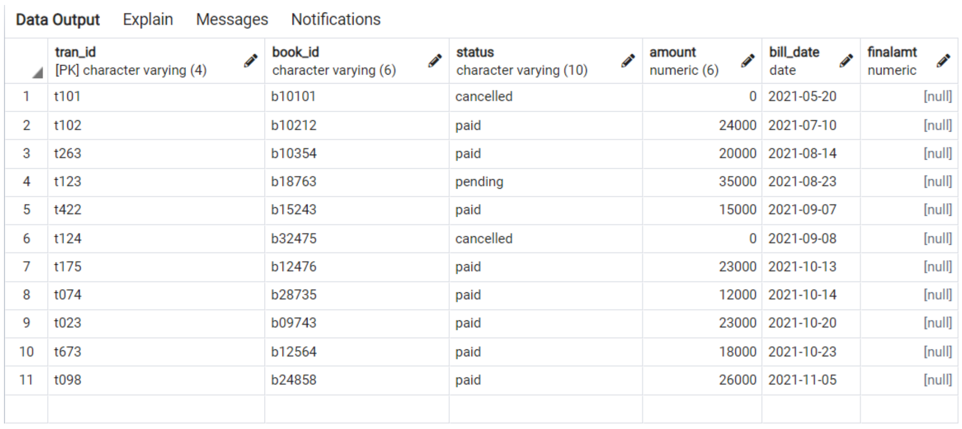
5.7 CURSOR

5.7.1 Create a cursor which adds GST amount of 5 percent to every record of billing table and updates it in the finalamt column.

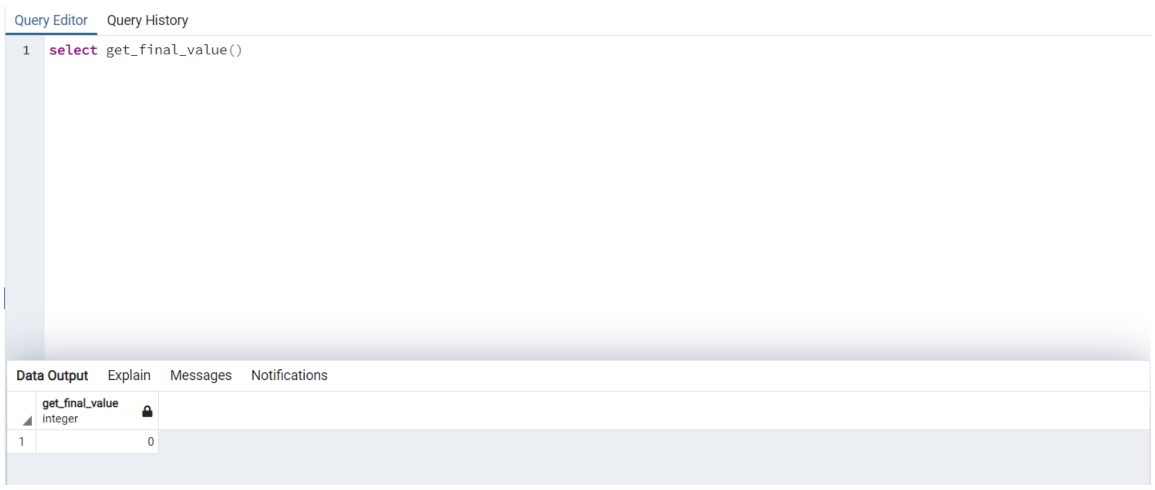
```
create or replace function get_final_value() returns int as $$
declare tranno int:=1;
 curid varchar(4);
c int;
icur cursor for select tran_id
from billing;
begin open icur;

c=count(*) from billing;

fetch next from icur into curid;

while c>0 loop update billing set finalamt=amount*0.05+amount where
tran_id=curid;
c=c-1;
fetch next from icur into curid; end
loop;
close icur; return
c; end; $$
language plpgsql
```

Query:

select get_final_value()



	tran_id [PK] character varying (4)	book_id character varying (6)	status character varying (10)	amount numeric (6)	bill_date date	finalamt numeric
1	t101	b10101	cancelled	0	2021-05-20	[null]
2	t102	b10212	paid	24000	2021-07-10	[null]
3	t263	b10354	paid	20000	2021-08-14	[null]
4	t123	b18763	pending	35000	2021-08-23	[null]
5	t422	b15243	paid	15000	2021-09-07	[null]
6	t124	b32475	cancelled	0	2021-09-08	[null]
7	t175	b12476	paid	23000	2021-10-13	[null]
8	t074	b28735	paid	12000	2021-10-14	[null]
9	t023	b09743	paid	23000	2021-10-20	[null]
10	t673	b12564	paid	18000	2021-10-23	[null]
11	t098	b24858	paid	26000	2021-11-05	[null]

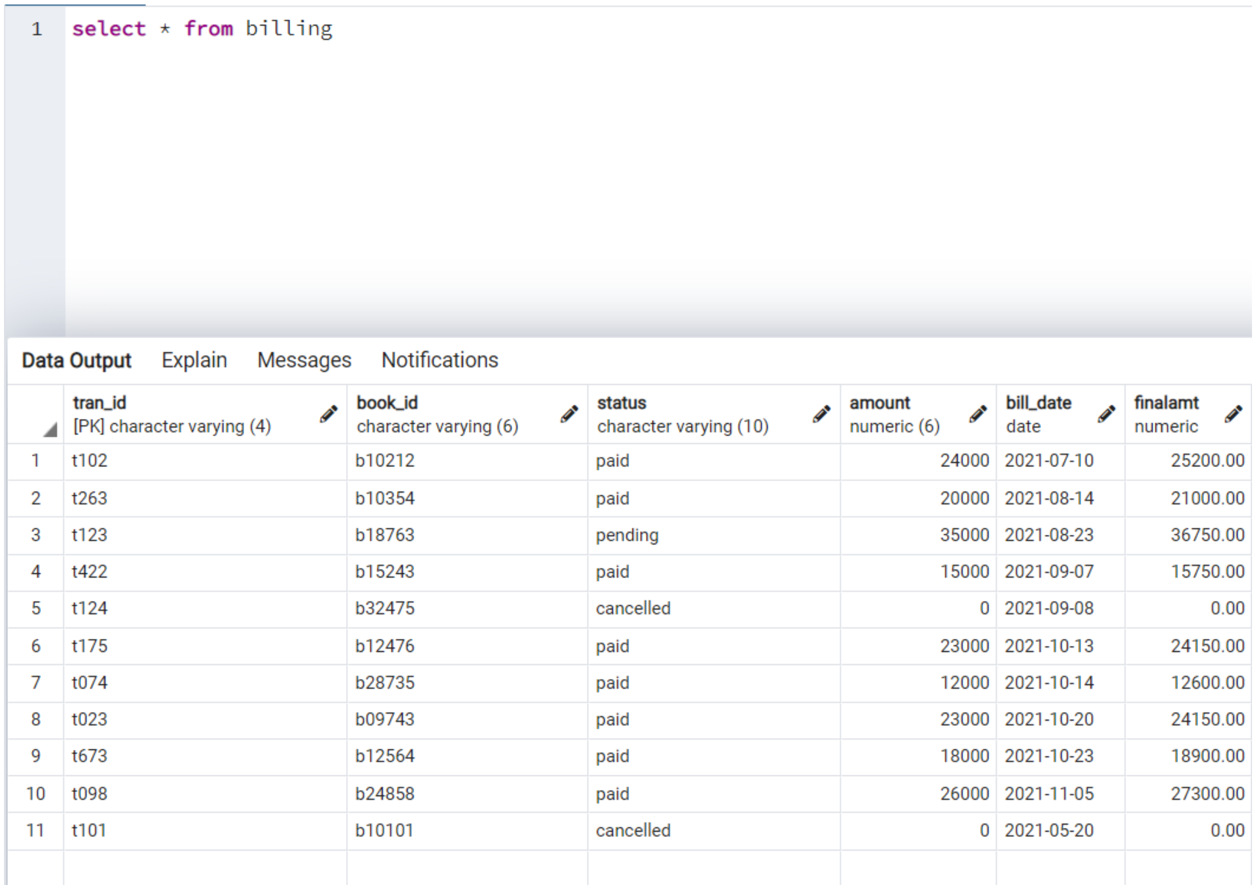


Query Editor Query History

```
select get_final_value()
```

Data Output Explain Messages Notifications

	get_final_value integer
1	0



```
select * from billing
```

Data Output Explain Messages Notifications

	tran_id [PK] character varying (4)	book_id character varying (6)	status character varying (10)	amount numeric (6)	bill_date date	finalamt numeric
1	t102	b10212	paid	24000	2021-07-10	25200.00
2	t263	b10354	paid	20000	2021-08-14	21000.00
3	t123	b18763	pending	35000	2021-08-23	36750.00
4	t422	b15243	paid	15000	2021-09-07	15750.00
5	t124	b32475	cancelled	0	2021-09-08	0.00
6	t175	b12476	paid	23000	2021-10-13	24150.00
7	t074	b28735	paid	12000	2021-10-14	12600.00
8	t023	b09743	paid	23000	2021-10-20	24150.00
9	t673	b12564	paid	18000	2021-10-23	18900.00
10	t098	b24858	paid	26000	2021-11-05	27300.00
11	t101	b10101	cancelled	0	2021-05-20	0.00

5.8 VIEW

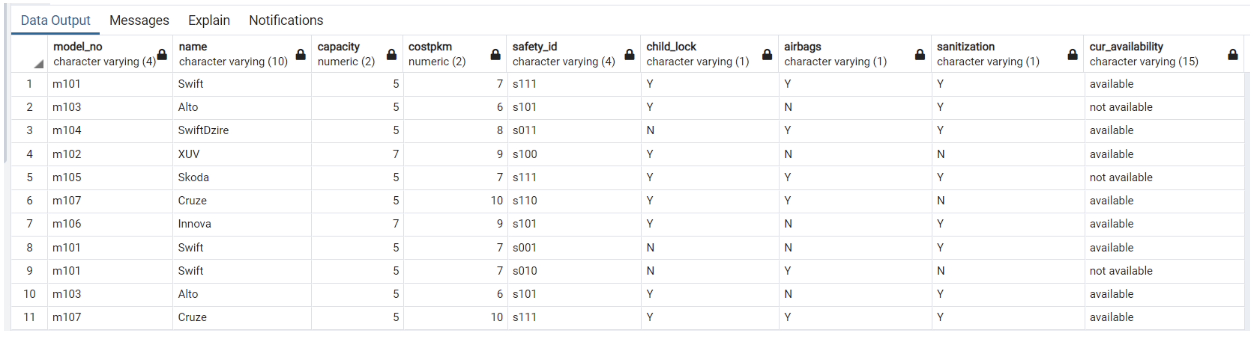
5.8.1 Display car info which includes car categories, safety measures and car availability using View .

CREATE VIEW carinfo AS SELECT
car_category.*,safety_measures.*,car.cur_availability FROM
car INNER JOIN car_category ON
car.model_no=car_category.model_no INNER JOIN
safety_measures ON
car.safety_id=safety_measures.safety_id;

QUERY:

SELECT * from carinfo;

OUTPUT:



	model_no character varying (4)	name character varying (10)	capacity numeric (2)	costpkm numeric (2)	safety_id character varying (4)	child_lock character varying (1)	airbags character varying (1)	sanitization character varying (1)	cur_availability character varying (15)
1	m101	Swift	5	7	s111	Y	Y	Y	available
2	m103	Alto	5	6	s101	Y	N	Y	not available
3	m104	SwiftDzire	5	8	s011	N	Y	Y	available
4	m102	XUV	7	9	s100	Y	N	N	available
5	m105	Skoda	5	7	s111	Y	Y	Y	not available
6	m107	Cruze	5	10	s110	Y	Y	N	available
7	m106	Innova	7	9	s101	Y	N	Y	available
8	m101	Swift	5	7	s001	N	N	Y	available
9	m101	Swift	5	7	s010	N	Y	N	not available
10	m103	Alto	5	6	s101	Y	N	Y	available
11	m107	Cruze	5	10	s111	Y	Y	Y	available

6. FUTURE ENHANCEMENTS OF THE SYSTEM

- We are planning to further develop the database based on the vehicle owner’s perspective.
- We will design Front-end Design in HTML, CSS, JavaScript and Develop Bank-end in Python.
- For security purposes, New Bookings will be done using OTP.
- Moreover, we are planning to add a tracking system which would help us locate the car for security purposes.
- Furthermore, we are planning to introduce late fee penalties in case someone fails to return the car at the designated time.
- With the addition of GUI features, it will be easier for both parties to register and make use of the application.
- We are planning to further enhance this database to make it capable of storing huge amounts of data with ease of access and retrieval.
- Moreover, we are also planning to add a complaint system in order to reach out to distressed customers.

7.BIBLIOGRAPHY

- For the successful implementation of this project, we referred to many websites and books.
- We created the ER Diagram on app.diagrams.net and Schema Diagram on "PostgreSQL."
- We also referred a lot of online material for syntax of procedures, triggers, cursors.

Reference book:
Data Base System Concepts
-Henry F. Korth & A. Silberschatz 2nd Ed. McGraw-Hill 1991

Reference Websites:

- https://www.stackoverflow.com/
- https://app.diagrams.net/?src=about
- https://www.w3school.com/
- https://www.tutorialspoint.com/
- http://www.mysqltutorial.org/