

DBMS - Mini Project

BIKE RENTAL MANAGEMENT SYSTEM



Submitted By:

Name: Pratik M Katti
SRN: PES1UG20CS577
V Semester Section – J

Short Description and Scope of the Project

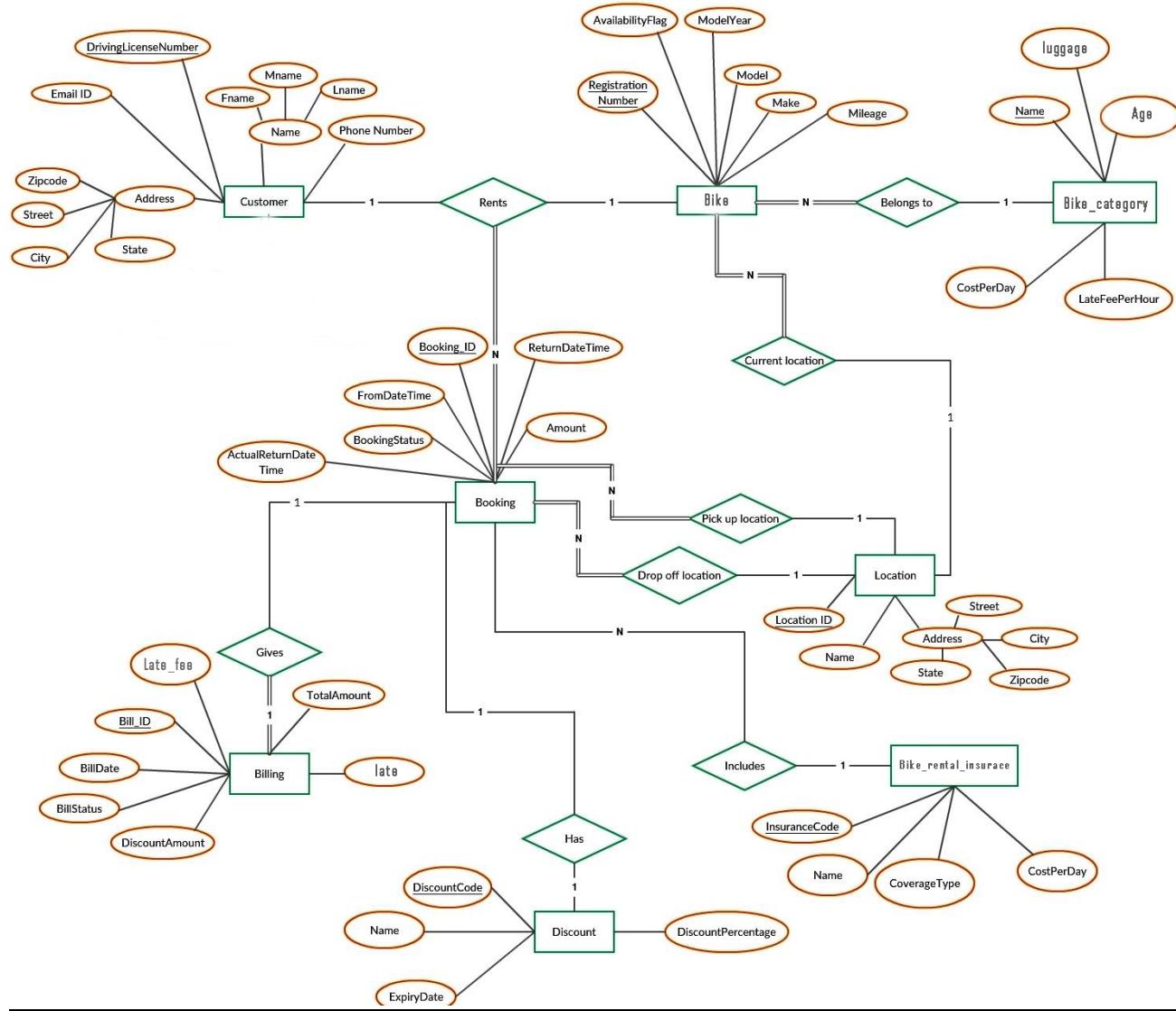
This system is designed to help the customers to take bikes or two-wheelers for rent.

In bike rental system, Customer can rent a Bike based on make,model and category. This project provides customer to have different pick-up and drop-off locations and will impose late fee if the rental car is returned beyond the return date and time.

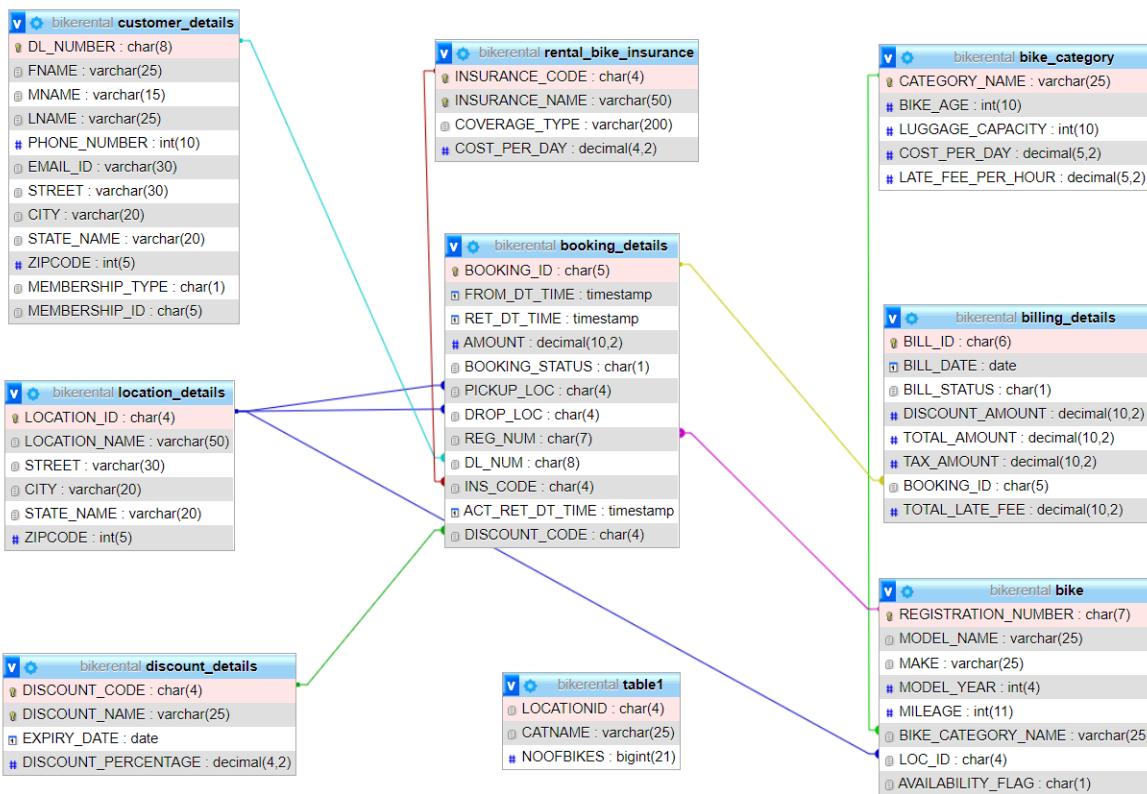
Entities

- Customer
- Bike
- Bike category
- Booking
- Billing
- Location
- Discount
- Bike Rental Insurance

ER diagram



Relational Schema



DDL statements - Building the database

1.CUSTOMER_DETAILS

```
CREATE TABLE CUSTOMER_DETAILS
( DL_NUMBER CHAR(8) NOT NULL,
FNAME VARCHAR(25) NOT NULL,
MNAME VARCHAR(15),
LNAME VARCHAR(25) NOT NULL,
PHONE_NUMBER INTEGER(10) NOT NULL,
EMAIL_ID VARCHAR(30) NOT NULL,
STREET VARCHAR(30) NOT NULL,
CITY VARCHAR(20) NOT NULL,
STATE_NAME VARCHAR(20) NOT NULL,
ZIPCODE INTEGER(5) NOT NULL,
MEMBERSHIP_TYPE CHAR(1) DEFAULT 'N' NOT NULL,
MEMBERSHIP_ID CHAR(5),
CONSTRAINT CUSTOMERPK
PRIMARY KEY (DL_NUMBER)
);
```

2.BIKE_CATEGORY

```
CREATE TABLE BIKE_CATEGORY
( CATEGORY_NAME VARCHAR(25) NOT NULL,
```

```
BIKE AGE INTEGER (10) NOT NULL,  
LUGGAGE_CAPACITY INTEGER (10) NOT NULL,  
COST_PER_DAY NUMERIC(5,2) NOT NULL,  
LATE_FEE_PER_HOUR NUMERIC(5,2) NOT NULL,  
CONSTRAINT BIKECATEGORYPK  
PRIMARY KEY (CATEGORY_NAME)  
);
```

3.LOCATION_DETAILS

```
CREATE TABLE LOCATION_DETAILS  
( LOCATION_ID CHAR(4) NOT NULL,  
LOCATION_NAME VARCHAR(50) NOT NULL,  
STREET VARCHAR(30) NOT NULL,  
CITY VARCHAR(20) NOT NULL,  
STATE_NAME VARCHAR(20) NOT NULL,  
ZIPCODE INTEGER(5) NOT NULL,  
CONSTRAINT LOCATIONPK  
PRIMARY KEY (LOCATION_ID)  
);
```

4. BIKE

```
CREATE TABLE BIKE  
( REGISTRATION_NUMBER CHAR(7) NOT NULL,  
MODEL_NAME VARCHAR(25) NOT NULL,  
MAKE VARCHAR(25) NOT NULL,
```

```
MODEL_YEAR INTEGER(4) NOT NULL,  
MILEAGE INTEGER NOT NULL,  
BIKE_CATEGORY_NAME VARCHAR(25) NOT NULL,  
LOC_ID CHAR(4) NOT NULL,  
AVAILABILITY_FLAG CHAR(1) NOT NULL,  
CONSTRAINT BIKEPK  
PRIMARY KEY (REGISTRATION_NUMBER),  
CONSTRAINT BIKEFK1  
FOREIGN KEY (BIKE_CATEGORY_NAME) REFERENCES BIKE_CATEGORY(CATEGORY_NAME),  
CONSTRAINT BIKEFK2  
FOREIGN KEY (LOC_ID) REFERENCES LOCATION_DETAILS(LOCATION_ID)  
);
```

5. DISCOUNT_DETAILS

```
CREATE TABLE DISCOUNT_DETAILS  
( DISCOUNT_CODE CHAR(4) NOT NULL,  
DISCOUNT_NAME VARCHAR(25) NOT NULL,  
EXPIRY_DATE DATE NOT NULL,  
DISCOUNT_PERCENTAGE NUMERIC(4,2) NOT NULL,  
CONSTRAINT DISCOUNTPK  
PRIMARY KEY (DISCOUNT_CODE),  
CONSTRAINT DISCOUNTSK  
UNIQUE (DISCOUNT_NAME)  
);
```

6.RENTAL_BIKE_INSURANCE

```
CREATE TABLE RENTAL_BIKE_INSURANCE
( INSURANCE_CODE CHAR(4) NOT NULL,
INSURANCE_NAME VARCHAR(50) NOT NULL,
COVERAGE_TYPE VARCHAR(200) NOT NULL,
COST_PER_DAY NUMERIC(4,2) NOT NULL,
CONSTRAINT INSURANCEPK
PRIMARY KEY (INSURANCE_CODE),
CONSTRAINT INSURANCESK
UNIQUE (INSURANCE_NAME)
);
```

7.BOOKING_DETAILS

```
CREATE TABLE BOOKING_DETAILS
( BOOKING_ID CHAR(5) NOT NULL,
FROM_DT_TIME TIMESTAMP NOT NULL,
RET_DT_TIME TIMESTAMP NOT NULL,
AMOUNT NUMERIC(10,2) NOT NULL,
BOOKING_STATUS CHAR(1) NOT NULL,
PICKUP_LOC CHAR(4) NOT NULL,
DROP_LOC CHAR(4) NOT NULL,
REG_NUM CHAR(7) NOT NULL,
DL_NUM CHAR(8) NOT NULL,
INS_CODE CHAR(4),
ACT_RET_DT_TIME TIMESTAMP,
DISCOUNT_CODE CHAR(4),
```

```
CONSTRAINT BOOKINGPK
PRIMARY KEY (BOOKING_ID),
CONSTRAINT BOOKINGFK1
FOREIGN KEY (PICKUP_LOC) REFERENCES LOCATION_DETAILS(LOCATION_ID),
CONSTRAINT BOOKINGFK2
FOREIGN KEY (DROP_LOC) REFERENCES LOCATION_DETAILS(LOCATION_ID),
CONSTRAINT BOOKINGFK3
FOREIGN KEY (REG_NUM) REFERENCES BIKE(REGISTRATION_NUMBER),
CONSTRAINT BOOKINGFK4
FOREIGN KEY (DL_NUM) REFERENCES CUSTOMER_DETAILS(DL_NUMBER),
CONSTRAINT BOOKINGFK5
FOREIGN KEY (INS_CODE) REFERENCES RENTAL_BIKE_INSURANCE(INSURANCE_CODE),
CONSTRAINT BOOKINGFK6
FOREIGN KEY (DISCOUNT_CODE) REFERENCES DISCOUNT_DETAILS(DISCOUNT_CODE)
);
```

8. BILLING_DETAILS

```
CREATE TABLE BILLING_DETAILS
( BILL_ID CHAR(6) NOT NULL,
BILL_DATE DATE NOT NULL,
BILL_STATUS CHAR(1) NOT NULL,
DISCOUNT_AMOUNT NUMERIC(10,2) NOT NULL,
TOTAL_AMOUNT NUMERIC(10,2) NOT NULL,
Late_fee NUMERIC(10,2) NOT NULL,
BOOKING_ID CHAR(5) NOT NULL,
Late CHAR NOT NULL,
```

```
CONSTRAINT BILLINGPK
PRIMARY KEY (BILL_ID),
CONSTRAINT BILLINGFK1
FOREIGN KEY (BOOKING_ID) REFERENCES BOOKING_DETAILS(BOOKING_ID)
);
```

Populating the Database

CUSTOMER_DETAILS

```
INSERT INTO CUSTOMER_DETAILS

VALUES('F1234554','KEERTHIKUMAR',NULL,'RAVICHANDRAN','4696004267','keerthi@gmail.com',
'700 CAMPBELL RD','RICHARDSON','TEXAS',75080,'M','M1001');

INSERT INTO CUSTOMER_DETAILS VALUES('F9764521', 'NIVEDITHA',NULL,'VARADHA
CHANDRASEKARAN','4696478596', 'nivi07@gmail.com','800 RENNER
RD','RICHARDSON','TEXAS',75080,'M','M1002');

INSERT INTO CUSTOMER_DETAILS VALUES('F2345611',
'SURESH','KUMAR','GOPALAKRISHNAN','8189187546', 'suresh2234@gmail.com','6547 CANOGA
AVE','CANOGA PARK','CALIFORNIA',91303,'N',NULL);

INSERT INTO CUSTOMER_DETAILS VALUES('R8763578', 'MARK',NULL,'HUFF','7345678902',
'markhuff@gmail.com','1445 ROSS AVE','DALLAS','TEXAS',75202,'N',NULL);

INSERT INTO CUSTOMER_DETAILS VALUES('I3478953', 'MARK','S','TOWNSEND','9872563478',
'markstown@gmail.com','7825 MCCALLUM BLVD','DALLAS','TEXAS',75252,'M','M1003');

INSERT INTO CUSTOMER_DETAILS VALUES('E7521097', 'MITA',NULL,'RANA','9098123429',
'mitarana@gmail.com','367 MEANDERING WAY','HOUSTON','TEXAS',76245,'N',NULL);

INSERT INTO CUSTOMER_DETAILS VALUES('T0981237', 'DANISH',NULL,'HASSAN','6712890345',
'danishhasan@gmail.com','888 PRESTON ROAD','DULLES','VIRGINIA',92367,'M','M1004');
```

```
INSERT INTO CUSTOMER_DETAILS VALUES('F0091266', 'MIKE',NULL,'BOYEAR','7892340918',
'mikeboy@gmail.com','1007 DALLAS PARKWAY','DALLAS','TEXAS',72212,'N',NULL);
INSERT INTO CUSTOMER_DETAILS VALUES('P1234567', 'CHRIS',NULL,'ALEXANDER','9902489',
'chrisalex@gmail.com','2256 WALL STREET','NEWARK','NEW JERSEY',65289,'M','M1005');
INSERT INTO CUSTOMER_DETAILS VALUES('V5690245', 'VELA','R','REYNALDO','9908762514',
'reyvela@gmail.com','0099 ALMA ROAD','DULLES','VIRGINIA',97325,'N',NULL);
```

BIKE_CATEGORY

```
INSERT INTO BIKE_CATEGORY VALUES('Scooter',2,5,30,0.9);
INSERT INTO BIKE_CATEGORY VALUES('Sport bike',3,5,32,0.96);
INSERT INTO BIKE_CATEGORY VALUES('Cruiser',3,5,35,1.05);
INSERT INTO BIKE_CATEGORY VALUES('Adventure tourer',3,5,38,1.14);
INSERT INTO BIKE_CATEGORY VALUES('Dirt bike',4,5,40,1.2);
INSERT INTO BIKE_CATEGORY VALUES('LUXURY BIKE',5,5,75,2.25);
INSERT INTO BIKE_CATEGORY VALUES('Cruiser Elite',2,5,36,1.08);
INSERT INTO BIKE_CATEGORY VALUES('mini bike',5,7,70,2.1);
```

LOCATION_DETAILS

```
INSERT INTO LOCATION_DETAILS VALUES
('L101','DALLAS LOVE FIELD AIRPORT','Herb KelleherWay','Dallas','Texas',75235);
INSERT INTO LOCATION_DETAILS VALUES('L102','LOS ANGELES INTL AIRPORT','World Way','Los
Angeles','California',90045);
INSERT INTO LOCATION_DETAILS VALUES('L103','DALLAS/ FORT WORTH INTL
AIRPORT','International Pkwy','DFW Airport','Texas',75261);
INSERT INTO LOCATION_DETAILS VALUES('L104','WEST HOUSTON AIRPORT','Groschke
Rd','Houston','Texas',77094);
INSERT INTO LOCATION_DETAILS VALUES('L105','WASHINGTON DULLES INTL AIRPORT','Saarinen
Cir','Dulles','Virginia',20226);
```

```
INSERT INTO LOCATION_DETAILS VALUES('L106','NEWARK LIBERTY INTL AIRPORT','Brewster Rd','Newark','New Jersey',07114);
INSERT INTO LOCATION_DETAILS VALUES('L107','SALT LAKE CITY INTL AIRPORT','N Terminal Dr','Salt Lake City','Utah',84122);
```

BIKE

```
INSERT INTO BIKE VALUES('ABX1234','ACTIVA','HONDA',2014,10000,'Scooter','L101','A');
INSERT INTO BIKE VALUES('SDF4567','VXL','VESPA',2015,15000,'Scooter','L102','N');
INSERT INTO BIKE VALUES('WER3245','ACCENT','TVS',2014,12356,'Scooter','L103','A');
INSERT INTO BIKE VALUES('GLZ2376','COROLLA','BAJAJ',2022,5000,'Scooter','L104','A');
INSERT INTO BIKE VALUES('HJK1234','ACTIVA','HONDA',2015,20145,'Scooter','L102','N');
INSERT INTO BIKE VALUES('GLS7625','FOCUS','VESPA',2014,12000,'Sport bike','L107','A');
INSERT INTO BIKE VALUES('FKD8202','GOLF','DUCATI',2022,9000,'Sport bike','L106','A');
INSERT INTO BIKE VALUES('HNX1890','PRIUS','BAJAJ',2015,15690,'Sport bike','L105','N');
INSERT INTO BIKE VALUES('KJS1983','PRIUS','BAJAJ',2014,20900,'Sport bike','L104','A');
INSERT INTO BIKE VALUES('SDL9356','FOCUS','VESPA',2022,10009,'Sport bike','L103','A');
INSERT INTO BIKE VALUES('OTY7293','CRUZE','ROYAL ENFIELD',2022,17800,'Cruiser','L102','A');
INSERT INTO BIKE VALUES('QWE4562','LEGACY','HARLEY',2012,13420,'Cruiser','L101','A');
INSERT INTO BIKE VALUES('CXZ2356','AVENGER','DODGE',2015,5000,'Cruiser','L102','A');
INSERT INTO BIKE VALUES('ASD9090','ACCORD','HONDA',2022,200,'Cruiser','L103','A');
INSERT INTO BIKE VALUES('UYT3981','LEGACY','HARLEY',2013,22750,'Cruiser','L104','A');
INSERT INTO BIKE VALUES('TRE9726','200','KTM',2012,14320,'Adventure tourer','L105','A');
INSERT INTO BIKE VALUES('HGF5628','TAURUS','VESPA',2013,15540,'Adventure tourer','L106','A');
INSERT INTO BIKE VALUES('LKJ7253','200','KTM',2014,22300,'Adventure tourer','L107','A');
INSERT INTO BIKE VALUES('VBN6283','TAURUS','VESPA',2015,17500,'Adventure tourer','L101','A');
INSERT INTO BIKE VALUES('POI7281','200','KTM',2022,18830,'Adventure tourer','L102','N');
INSERT INTO BIKE VALUES('MNB8654','FALCON','VESPA',2012,10900,'Dirt bike','L103','A');
```

```

INSERT INTO BIKE VALUES('UHV9786', 'IMPALA', 'ROYAL ENFIELD', 2013, 11500, 'Dirt
bike', 'L104', 'A');
INSERT INTO BIKE VALUES('EFB5427', 'WAYFARER', 'VESPA', 2014, 14350, 'Dirt
bike', 'L105', 'A');
INSERT INTO BIKE VALUES('PLM9873', 'IMPALA', 'ROYAL ENFIELD', 2015, 18900, 'Dirt
bike', 'L106', 'A');
INSERT INTO BIKE VALUES('WDV2458', 'FALCON', 'VESPA', 2022, 5600, 'Dirt bike', 'L107', 'A');
INSERT INTO BIKE VALUES('QSC8709', 'MKZ', 'HAYABUSA', 2012, 18700, 'LUXURY
BIKE', 'L101', 'A');
INSERT INTO BIKE VALUES('TGB8961', 'GENESIS', 'TVS', 2013, 17620, 'LUXURY BIKE', 'L102', 'A');
INSERT INTO BIKE VALUES('MKU0172', 'TLX', 'ACURA', 2014, 12345, 'LUXURY BIKE', 'L103', 'A');
INSERT INTO BIKE VALUES('CFT1908', '328I', 'BMW', 2015, 10800, 'LUXURY BIKE', 'L104', 'A');
INSERT INTO BIKE VALUES('WHM7619', 'AVALON', 'BAJAJ', 2022, 7800, 'LUXURY BIKE', 'L105', 'A');
INSERT INTO BIKE VALUES('WLZ8955', 'ESCAPE', 'VESPA', 2012, 19800, 'Cruiser
Elite', 'L106', 'A');
INSERT INTO BIKE VALUES('QI07621', 'EQUINOX', 'ROYAL ENFIELD', 2013, 17560, 'Cruiser
Elite', 'L107', 'A');
INSERT INTO BIKE VALUES('YSN1927', 'PATHFINDER', 'AVENGER', 2014, 14390, 'Cruiser
Elite', 'L101', 'A');
INSERT INTO BIKE VALUES('EDM8610', 'GLA', 'HARLEY', 2015, 12900, 'Cruiser
Elite', 'L102', 'A');
INSERT INTO BIKE VALUES('AHK7325', 'RAV4', 'BAJAJ', 2022, 3400, 'Cruiser Elite', 'L103', 'A');
INSERT INTO BIKE VALUES('SHK7767', 'QUEST', 'AVENGER', 2012, 23478, 'mini bike', 'L107', 'A');
INSERT INTO BIKE VALUES('JSL7920', 'ODYSSEY', 'HONDA', 2013, 19320, 'mini bike', 'L106', 'A');
INSERT INTO BIKE VALUES('PAJ5289', 'GRAND BIKEAVAN', 'DODGE', 2014, 23478, 'mini
bike', 'L105', 'A');
INSERT INTO BIKE VALUES('TSJ6290', 'QUEST', 'AVENGER', 2015, 13200, 'mini bike', 'L104', 'A');
INSERT INTO BIKE VALUES('MW09296', 'ODYSSEY', 'HONDA', 2022, 2300, 'mini bike', 'L103', 'A');

```

DISCOUNT_DETAILS

```

INSERT INTO DISCOUNT_DETAILS VALUES ('D678', 'IBM CORPORATE', '2018-01-25', 25);
INSERT INTO DISCOUNT_DETAILS VALUES ('D234', 'CTS CORPORATE', '2019-09-02', 20);
INSERT INTO DISCOUNT_DETAILS VALUES ('D756', 'HOLIDAY SPECIAL', '2017-10-29', 10);

```

```
INSERT INTO DISCOUNT_DETAILS VALUES ('D109','WEEKLY RENTALS','2020-11-09',25);
INSERT INTO DISCOUNT_DETAILS VALUES ('D972','ONE WAY SPECIAL','2022-12-15',20);
INSERT INTO DISCOUNT_DETAILS VALUES ('D297','UPGRADE SPECIAL','2018-02-18',20);
```

RENTAL_BIKE_INSURANCE

```
INSERT INTO RENTAL_BIKE_INSURANCE VALUES
('I201', 'COLLISION DAMAGE WAIVER', 'Covers theft and total damage to the rental
BIKE',15);
INSERT INTO RENTAL_BIKE_INSURANCE VALUES('I202', 'SUPPLEMENTAL LIABILITY PROTECTION',
'Covers damage done to others',12);
INSERT INTO RENTAL_BIKE_INSURANCE VALUES('I203', 'PERSONAL ACCIDENT INSURANCE', 'Covers
medical costs for driver and passengers',10);
INSERT INTO RENTAL_BIKE_INSURANCE VALUES('I204', 'PERSONAL EFFECTS COVERAGE', 'Covers
theft of personal belongings',5);
```

BOOKING_DETAILS

```
INSERT INTO BOOKING_DETAILS VALUES
('B1001','2022-01-20 10:00:00','2022-01-25
10:00:00',150,'R','L101','L101','ABX1234','F12345546',NULL,'2022-01-25
10:00:00','D756');
INSERT INTO BOOKING_DETAILS VALUES('B1002','2022-01-21 11:00:00','2022-01-24
10:00:00',90,'C','L102','L102','SDF4567','T0981237',NULL,NULL,NULL);
INSERT INTO BOOKING_DETAILS VALUES('B1003','2022-02-10 13:00:00','2022-01-15 13:00:00'
,450,'R','L101','L101','QSC8709','R8763578','I201','2022-01-15 13:00:00',NULL);
```

```
INSERT INTO BOOKING_DETAILS VALUES('B1004','2022-04-24 13:00:00','2022-04-25  
20:30:00',48,'R','L106','L106','WLZ8955','F0091266','I202','2022-04-23 20:30:00'  
, 'D234');  
INSERT INTO BOOKING_DETAILS VALUES('B1005','2022-04-18 09:00:00','2022-04-25 09:00:00'  
,266,'B','L102','L106','POI7281','P1234567',NULL,NULL,'D972');  
INSERT INTO BOOKING_DETAILS VALUES('B1006','2022-04-21 17:00:00','2022-04-25 17:00:00'  
,228,'B','L105','L107','HNX1890','V5690245','I203',NULL,'D234');  
INSERT INTO BOOKING_DETAILS VALUES('B1007','2022-04-22 08:00:00' , '2022-04-25  
17:00:00',405,'B','L102','L102','SDF4567','I3478953','I201',NULL,'D756');  
INSERT INTO BOOKING_DETAILS VALUES('B1008','2022-04-11 08:00:00' , '2022-04-25 17:00:00'  
,630,'B','L102','L102','HJK1234','T0981237','I201',NULL,'D756');
```

BILLING_DETAILS

```
INSERT INTO BILLING_DETAILS VALUES('BL1001','2022-01-25','P',24.36 ,138.03,12.38  
, 'B1001',N);  
INSERT INTO BILLING_DETAILS VALUES('BL1002','2022-01-15','P',0 ,487.13 ,12.38  
, 'B1003',N);  
INSERT INTO BILLING_DETAILS VALUES('BL1003','2022-04-24','P',10.39 ,41.57 ,3.96  
, 'B1004',N);
```

Queries

JOIN Queries

1.SQL Query to displays bike model name and cost per day of that bike.

Query

```
SELECT DISTINCT MODEL_NAME, COST_PER_DAY FROM BIKE_CATEGORY CC INNER  
JOIN BIKE C ON CC.CATEGORY_NAME = C.BIKE_CATEGORY_NAME;
```

Showing rows 0 - 24 (29 total, Query took 0.0047 seconds.)

```
SELECT DISTINCT MODEL_NAME,COST_PER_DAY FROM BIKE_CATEGORY CC INNER JOIN BIKE C ON CC.CATEGORY_NAME = C.BIKE_CATEGORY_NAME;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 > >> | Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

MODEL_NAME	COST_PER_DAY
TAURUS	38
200	38
ACCORD	35
AVENGER	35
CRUZE	35
LEGACY	35
RAV4	36
GLA	36
EQUINOX	36
ESCAPE	36
PATHFINDER	36
WAYFARER	40
FALCON	40
IMPALA	40
328I	75
TLX	75
MKZ	75
GENESIS	75
AVALON	75
ODYSSEY	70
GRAND BIKEAVAN	70
QUEST	70
ACTIVA	30
COROLLA	30
VXL	30

2.SQL query to display bike name and its pickup location

Query

```
SELECT MAKE, MODEL_NAME ,LOCATION_NAME FROM BIKE B inner join location_details l on  
B.LOC_ID=l.LOCATION_ID;
```

```
SELECT MAKE, MODEL_NAME ,LOCATION_NAME FROM BIKE B inner join location_details l on B.LOC_ID=l.LOCATION_ID;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 ▾

> >>

|

Show all

Number of rows:

25 ▾

Filter rows:

Search this table

Sort by key:

None

+ Options

MAKE	MODEL_NAME	LOCATION_NAME
HAYABUSA	MKZ	DALLAS LOVE FIELD AIRPORT
HARLEY	LEGACY	DALLAS LOVE FIELD AIRPORT
VESPA	TAURUS	DALLAS LOVE FIELD AIRPORT
AVENGER	PATHFINDER	DALLAS LOVE FIELD AIRPORT
HONDA	ACTIVA	LOS ANGELES INTL AIRPORT
ROYAL ENFIELD	CRUZE	LOS ANGELES INTL AIRPORT
KTM	200	LOS ANGELES INTL AIRPORT
VESPA	VXL	LOS ANGELES INTL AIRPORT
TVS	GENESIS	LOS ANGELES INTL AIRPORT
HONDA	ACCORD	DALLAS/ FORT WORTH INTL AIRPORT
ACURA	TLX	DALLAS/ FORT WORTH INTL AIRPORT
VESPA	FALCON	DALLAS/ FORT WORTH INTL AIRPORT
HONDA	ODYSSEY	DALLAS/ FORT WORTH INTL AIRPORT
VESPA	FOCUS	DALLAS/ FORT WORTH INTL AIRPORT
TVS	ACCENT	DALLAS/ FORT WORTH INTL AIRPORT
BAJAJ	COROLLA	WEST HOUSTON AIRPORT
BAJAJ	PRIUS	WEST HOUSTON AIRPORT
AVENGER	QUEST	WEST HOUSTON AIRPORT
ROYAL ENFIELD	IMPALA	WEST HOUSTON AIRPORT
HARLEY	LEGACY	WEST HOUSTON AIRPORT
VESPA	WAYFARER	WASHINGTON DULLES INTL AIRPORT
BAJAJ	PRIUS	WASHINGTON DULLES INTL AIRPORT
DODGE	GRAND BIKEAVAN	WASHINGTON DULLES INTL AIRPORT
KTM	200	WASHINGTON DULLES INTL AIRPORT
BAJAJ	AVALON	WASHINGTON DULLES INTL AIRPORT

3. Query to display the booked bikes with their booking ids

Query

```
SELECT MAKE, MODEL_NAME, BOOKING_ID FROM BIKE B JOIN booking_details C WHERE  
B.REGISTRATION_NUMBER=C.REG_NUM;
```

Showing rows 0 - 7 (8 total, Query took 0.0056 seconds.)

```
SELECT MAKE, MODEL_NAME, BOOKING_ID FROM BIKE B JOIN booking_details C WHERE B.REGISTRATION_NUMBER=C.REG_NUM;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows:

25 ▾

Filter rows:

Search this table

Sort by key:

None

+ Options

MAKE	MODEL_NAME	BOOKING_ID
HONDA	ACTIVA	B1001
VESPA	VXL	B1002
HAYABUSA	MKZ	B1003
VESPA	ESCAPE	B1004
KTM	200	B1005
BAJAJ	PRIUS	B1006
VESPA	VXL	B1007
HONDA	ACTIVA	B1008

4) SQL query to display bike name and their respective late fee per hour

Query

```
SELECT DISTINCT MAKE, MODEL_NAME, LATE_FEE_PER_HOUR FROM BIKE B JOIN BIKE_CATEGORY C  
WHERE B.BIKE_CATEGORY_NAME=C.CATEGORY_NAME
```

```
SELECT DISTINCT MAKE, MODEL_NAME, LATE_FEE_PER_HOUR FROM BIKE B JOIN BIKE_CATEGORY C WHERE B.BIKE_CATEGORY_NAME=C.CATEGORY_NAME;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

1 ▾

> ▾

>> ▾

| ▾

Show all

| ▾

Number of rows:

25 ▾

Filter rows: Search this table

Sort by key: None ▾

+ Options

MAKE	MODEL_NAME	LATE_FEE_PER_HOUR
VESPA	TAURUS	1
KTM	200	1
HONDA	ACCORD	1
DODGE	AVENGER	1
ROYAL ENFIELD	CRUZE	1
HARLEY	LEGACY	1
BAJAJ	RAV4	1
HARLEY	GLA	1
ROYAL ENFIELD	EQUINOX	1
VESPA	ESCAPE	1
AVENGER	PATHFINDER	1
VESPA	WAYFARER	1
VESPA	FALCON	1
ROYAL ENFIELD	IMPALA	1
BMW	328I	2
ACURA	TLX	2
HAYABUSA	MKZ	2
TVS	GENESIS	2
BAJAJ	AVALON	2
HONDA	ODYSSEY	2
DODGE	GRAND BIKEAVAN	2
AVENGER	QUEST	2
HONDA	ACTIVA	1
BAJAJ	COROLLA	1
VESPA	VXL	1

AGGREGATE Function

1. query to display bike category and number of bikes in each category

Query

```
SELECT BIKE_CATEGORY_NAME, COUNT(BIKE_CATEGORY_NAME) FROM BIKE GROUP BY BIKE_CATEGORY_NAME;
```

✓ Showing rows 0 - 7 (8 total, Query took 0.0036 seconds.)

```
SELECT BIKE_CATEGORY_NAME, COUNT(BIKE_CATEGORY_NAME) FROM BIKE GROUP BY BIKE_CATEGORY_NAME;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all | Number of rows: 25 ▾ Filter rows: Search this table

+ Options

BIKE_CATEGORY_NAME	COUNT(BIKE_CATEGORY_NAME)
Adventure tourer	5
Cruiser	5
Cruiser Elite	5
Dirt bike	5
LUXURY BIKE	5
mini bike	5
Scooter	5
Sport bike	5

2) displays total amount earned from renting bikes

Query

```
SELECT SUM(AMOUNT) FROM booking_details;
```

✓ Showing rows 0 - 0 (1 total, Query took 0.0025 seconds.)

```
SELECT SUM(AMOUNT) FROM booking_details;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all

Number of rows:

25 ▾

Filter rows:

Search this table

+ Options

SUM(AMOUNT)

2267.00

3) displays locations and number of bikes in that location

Query

```
SELECT LOCATION_NAME, COUNT(LOCATION_NAME) FROM location_details L JOIN BIKE B  
WHERE L.LOCATION_ID=B.LOC_ID GROUP BY LOCATION_NAME;
```

✓ Showing rows 0 - 6 (7 total, Query took 0.0061 seconds.)

```
SELECT LOCATION_NAME, COUNT(LOCATION_NAME) FROM location_details L JOIN BIKE B WHERE L.LOCATION_ID=B.LOC_ID GROUP BY LOCATION_NAME;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all

Number of rows:

25 ▾

Filter rows: Search this table

+ Options

LOCATION_NAME	COUNT(LOCATION_NAME)
DALLAS LOVE FIELD AIRPORT	4
DALLAS/ FORT WORTH INTL AIRPORT	6
LOS ANGELES INTL AIRPORT	5
NEWARK LIBERTY INTL AIRPORT	4
SALT LAKE CITY INTL AIRPORT	5
WASHINGTON DULLES INTL AIRPORT	5
WEST HOUSTON AIRPORT	5

4) displays the average amount earned from every booking

Query

```
SELECT AVG(AMOUNT) FROM booking_details;
```

Showing rows 0 - 0 (1 total, Query took 0.0031 seconds.)

```
SELECT AVG(AMOUNT) FROM booking_details;
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all

Number of rows:

25 ▾

Filter rows:

Search this table

+ Options

AVG(AMOUNT)

283.375000

SET Operations

1. retrieves bikes which are either in location L104 or L107

Query

```
(select * FROM bike where LOC_ID="L104" )union(SELECT * from bike where loc_id="L107");
```

Showing rows 0 - 8 (9 total, Query took 0.0064 seconds.)

```
(select * FROM bike where LOC_ID="L104" )union(SELECT * from bike where loc_id="L107");
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

REGISTRATION_NUMBER	MODEL_NAME	MAKE	MODEL_YEAR	MILEAGE	BIKE_CATEGORY_NAME	LOC_ID	AVAILABILITY_FLAG
GLZ2376	COROLLA	BAJAJ	2022	5000	Scooter	L104	A
KJS1983	PRIUS	BAJAJ	2014	20900	Sport bike	L104	A
TSJ6290	QUEST	AVENGER	2015	13200	mini bike	L104	A
UHV9786	IMPALA	ROYAL ENFIELD	2013	11500	Dirt bike	L104	A
UYT3981	LEGACY	HARLEY	2013	22750	Cruiser	L104	A
LKJ7253	200	KTM	2014	22300	Adventure tourer	L107	A
QIO7621	EQUINOX	ROYAL ENFIELD	2013	17560	Cruiser Elite	L107	A
SHK7767	QUEST	AVENGER	2012	23478	mini bike	L107	A
WDV2458	FALCON	VESPA	2022	5600	Dirt bike	L107	A

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

2. Displays the bills which are not cancels

Query

```
Select* from booking_details where booking_status="R")union (select * form booking_details where booking_status ="B")
```

Showing rows 0 - 7 (8 total, Query took 0.0034 seconds.)

```
(select * FROM BOOKING_DETAILS where BOOKING_STATUS="R" )union(select * FROM BOOKING_DETAILS where BOOKING_STATUS="B");
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

BOOKING_ID	FROM_DT_TIME	RET_DT_TIME	AMOUNT	BOOKING_STATUS	PICKUP_LOC	DROP_LOC	REG_NUM	DL_NUM	INS_CODE	ACT_RET_DT_TIME	DISCOUNT_CODE
B1001	2022-01-20	2022-01-25	150.00	R	L101	L101	ABX1234	F1234554	NULL	2022-01-25 10:00:00	D756
B1003	2022-02-10	2022-01-15	450.00	R	L101	L101	QSC8709	R8763578	I201	2022-01-15 13:00:00	NULL
B1004	2022-04-24	2022-04-25	48.00	R	L106	L106	WLZ8955	F0091266	I202	2022-04-23 20:30:00	D234
999	2022-11-23	2022-11-23	NULL	B	NULL	NULL	CXZ2356	NULL	NULL	2022-11-23 18:38:29	NULL
B1005	2022-04-18	2022-04-25	266.00	B	L102	L106	POI7281	P1234567	NULL	NULL	D972
B1006	2022-04-21	2022-04-25	228.00	B	L105	L107	HNX1890	V5690245	I203	NULL	D234
B1007	2022-04-22	2022-04-25	405.00	B	L102	L102	SDF4567	I3478953	I201	NULL	D756
B1008	2022-04-11	2022-04-25	630.00	B	L102	L102	HJK1234	T0981237	I201	NULL	D756

3. Retrieves the coupons whose discount percent is greater than 10% and expiry date is greater than 2017/01/15

Query

```
(SELECT * FROM discount_details WHERE DISCOUNT_PERCENTAGE>10)INTERSECT (SELECT *FROM discount_details WHERE EXPIRY_DATE>2017-01-15);
```

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

+ Options

DISCOUNT_CODE	DISCOUNT_NAME	EXPIRY_DATE	DISCOUNT_PERCENTAGE
D109	WEEKLY RENTALS	2022-11-16	25
D234	CTS CORPORATE	2023-09-14	20
D297	UPGRADE SPECIAL	2018-02-18	20
D678	IBM CORPORATE	2018-01-25	25
D972	ONE WAY SPECIAL	2022-12-15	20

4. displays bills which are lately paid

Query

```
(SELECT * FROM billing_details WHERE BILL_STATUS="P")INTERSECT (SELECT *FROM billing_details WHERE LATE="N");
```

Showing rows 0 - 1 (2 total, Query took 0.0027 seconds.)

```
(SELECT * FROM billing_details WHERE BILL_STATUS='P') INTERSECT (SELECT * FROM billing_details WHERE LATE='N');
```

[Edit inline] [Edit] [Create PHP code]

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

BILL_ID	BILL_DATE	BILL_STATUS	DISCOUNT_AMOUNT	TOTAL_AMOUNT	LATE_FEE	BOOKING_ID	LATE
BL1002	2022-01-15	P	0.00	487.13	7.00	B1003	N
BL1003	2022-04-24	P	10.39	41.57	0.00	B1004	N

Functions

1.total_amount

This function takes registration number, booked date and return_date as parameter and returns Rental_Amount.

Amount is calculated using cost_per_day of the bike and number of days for which bike is rented
 Cost_per_day of the bike is obtained from its registration number and number of days is obtained by subtracting return date with booked date

```
DELIMITER $$

CREATE FUNCTION `total_amount`(`regno` VARCHAR(11), `booked_date` DATE,
`ReturnDateTime` DATE) RETURNS int(11)

BEGIN
DECLARE amt int;
DECLARE amount int;

SELECT COST_PER_DAY INTO amt FROM BIKE_CATEGORY CC INNER JOIN BIKE C ON
CC.CATEGORY_NAME = C.BIKE_CATEGORY_NAME WHERE C.REGISTRATION_NUMBER = regno;
```

```

set @amount=amt * datediff(booked_date,ReturnDateTime);

return amt;

END$$

DELIMITER ;

```

 MySQL returned an empty result set (i.e. zero rows). (Query took 0.0219 seconds.)

```
CREATE FUNCTION `total_amount`(`regno` VARCHAR(11), `booked_date` DATE, `ReturnDateTime` DATE) RETURNS int(11) BEGIN DECLARE amt int; DECLARE amount int; SELECT COST_PER_DAY INTO amt FROM BIKE_CATEGORY CC INNER JOIN BIKE C ON CC.CATEGORY_NAME = C.BIKE_CATEGORY_NAME WHERE C.REGISTRATION_NUMBER = regno; set @amount=amt * datediff(booked_date,ReturnDateTime); return amt; END;
```

[Edit inline] [Edit] [Create PHP code]

Calling function total_amount() with appropriate parameters

 Showing rows 0 - 0 (1 total, Query took 0.0021 seconds.)

```
SELECT total_amount("ABX1234","2022-01-20","2022-01-27");
```

Profiling [Edit inline] [Edit] [Explain SQL] [Create PHP code] [Refresh]

Show all

Number of rows:

25

Filter rows:

Search this table

+ Options

```
total_amount("ABX1234","2022-01-20","2022-01-27")
```

30

Stored Procedure

1.Late fee

This procedure updates late fee and status (late or not) in the billing_details table. It takes actual_returned_date, return_date, bill_number and register number as parameter .

This procedure checks whether actual_returned_date > return_date , if this condition is true then it calculates late fee.

Late fee is calculated with the help of late_fee_per_day which is fetched using register of that particular bike

```
DELIMITER $$  
CREATE PROCEDURE `LATE_FEE`(IN `actualReturnDateTime` TIMESTAMP, IN  
`ReturnDateTime` TIMESTAMP, IN `BILLNO` VARCHAR(11), IN `regNum` VARCHAR(11))  
  
BEGIN  
  
DECLARE lateFeePerHour INT ;  
DECLARE hourDifference DECIMAL(10,2);  
  
SELECT LATE_FEE_PER_HOUR INTO lateFeePerHour FROM BIKE_CATEGORY AS CC INNER  
JOIN BIKE AS C ON CC.CATEGORY_NAME = C.BIKE_CATEGORY_NAME WHERE  
C.REGISTRATION_NUMBER = regNum;  
IF actualReturnDateTime > ReturnDateTime THEN SELECT  
DATEDIFF(actualReturnDateTime, ReturnDateTime) into hourDifference;  
  
CALL LATE(actualReturnDateTime, ReturnDateTime,BILLNO);  
UPDATE billing_details SET LATE_FEE=(hourDifference * lateFeePerHour) WHERE  
BILL_ID=BILLNO;  
  
END IF;  
  
END$$  
DELIMITER ;
```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0242 seconds.)

```
CREATE PROCEDURE `LATE_FEE`(IN `actualReturnDateTime` TIMESTAMP, IN `ReturnDateTime` TIMESTAMP, IN `BILLNO` VARCHAR(11), IN `regNum` VARCHAR(11)`) BEGIN DECLARE lateFeePerHour INT; DECLARE hourDifference DECIMAL(10,2); SELECT LATE_FEE_PER_HOUR INTO lateFeePerHour FROM BIKE_CATEGORY AS CC INNER JOIN BIKE AS C ON CC.CATEGORY_NAME = C.BIKE_CATEGORY_NAME WHERE C.REGISTRATION_NUMBER = regNum; IF actualReturnDateTime > ReturnDateTime THEN SELECT DATEDIFF(actualReturnDateTime, ReturnDateTime) into hourDifference; END IF; CALL LATE(actualReturnDateTime, ReturnDateTime,BILLNO); UPDATE billing_details SET LATE_FEE=(hourDifference * lateFeePerHour) WHERE BILL_ID=BILLNO; END;
```

[Edit inline] [Edit] [Create PHP code]

Before calling LATE_FEE

+ Options									
	← T →	▼ BILL_ID	BILL_DATE	BILL_STATUS	DISCOUNT_AMOUNT	TOTAL_AMOUNT	LATE_FEE	BOOKING_ID	LATE
<input type="checkbox"/>				BL1001	2022-01-25	P	24.36	138.03	0.00
<input type="checkbox"/>				BL1002	2022-01-15	P	0.00	487.13	0.00
<input type="checkbox"/>				BL1003	2022-04-24	P	10.39	41.57	0.00

calling LATE_FEE

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0023 seconds.)

```
CALL `LATE_FEE`('2022-01-27 10:00:00', '2022-01-20 10:00:00', 'BL1001', 'ABX1234');
```

[Edit inline] [Edit] [Create PHP code]

After calling LATE_FEE

+ Options									
	← T →	▼ BILL_ID	BILL_DATE	BILL_STATUS	DISCOUNT_AMOUNT	TOTAL_AMOUNT	LATE_FEE	BOOKING_ID	LATE
<input type="checkbox"/>				BL1001	2022-01-25	P	24.36	138.03	7.00
<input type="checkbox"/>				BL1002	2022-01-15	P	0.00	487.13	0.00
<input type="checkbox"/>				BL1003	2022-04-24	P	10.39	41.57	0.00

2.updateamount

This procedure takes registration_number , booked_date, return_date and bill_id as parameters . It updates the amount in booking_details table.

When a bike is booked , This procedure is called and this procedure calls total_amount function.

```
DELIMITER $$  
CREATE PROCEDURE `updateamount`(IN `regno` VARCHAR(11), IN `booked_date` DATE,  
IN `ReturnDateTime` DATE, IN `billid` VARCHAR(11)`)  
  
BEGIN  
DECLARE amt int;
```

```

SELECT total_amount(regno,booked_date,ReturnDateTime) into amt;

UPDATE booking_details SET AMOUNT=amt WHERE BOOKING_ID = billid;

END$$

DELIMITER ;

```

MySQL returned an empty result set (i.e. zero rows). (Query took 0.0188 seconds.)

```

CREATE PROCEDURE `updateamount`(`IN` `regno` `VARCHAR(11)`, `IN` `booked_date` `DATE`, `IN` `ReturnDateTime` `DATE`, `IN` `billid` `VARCHAR(11)` ) BEGIN DECLARE amt int; SELECT total_amount(regno,booked_date,ReturnDateTime) into amt; UPDATE booking_details SET AMOUNT=amt WHERE BOOKING_ID = billid; END;

```

[\[Edit inline \]](#) [\[Edit \]](#) [\[Create PHP code \]](#)

Before calling updateamount

Showing rows 0 - 8 (9 total, Query took 0.0015 seconds.)

```

SELECT * FROM `booking_details`;

```

Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

	Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None					
+ Options	Edit Copy Delete	999	2022-11-23	2022-11-23	NULL	C	NULL	NULL	CXZ2356	NULL	NULL	2022-11-23 23:18:44
	Edit Copy Delete	B1001	2022-01-20	2022-01-25	150.00	R	L101	L101	ABX1234	F1234554	NULL	2022-01-25 10:00:00

Calling updateamount

Your SQL query has been executed successfully.

0 rows affected by the last statement inside the procedure.

```

SET @p0='ABX1234'; SET @p1='2022-01-20'; SET @p2='2022-01-27 10:00:0'; SET @p3='999'; CALL `updateamount`(@p0, @p1, @p2, @p3);

```

after calling updateamount

Showing rows 0 - 8 (9 total, Query took 0.0011 seconds.)

```

SELECT * FROM `booking_details`;

```

Profiling [\[Edit inline \]](#) [\[Edit \]](#) [\[Explain SQL \]](#) [\[Create PHP code \]](#) [\[Refresh \]](#)

	Show all	Number of rows:	25	Filter rows:	Search this table	Sort by key:	None					
+ Options	Edit Copy Delete	999	2022-11-23	2022-11-23	30.00	C	NULL	NULL	CXZ2356	NULL	NULL	2022-11-23 23:50:44

Triggers

1.BOOK_BIKE

Book_bike trigger is used to set the availability_flag of a bike to “not available (N)” when is bike is booked.

when a new bike is booked , Details of the booking are inserted in booking_details table , this activates book_bike trigger

```
CREATE TRIGGER `BOOK_BIKE` AFTER INSERT ON `booking_details`  
FOR EACH ROW BEGIN  
  
UPDATE BIKE SET AVAILABILITY_FLAG = 'N' WHERE REGISTRATION_NUMBER =  
NEW.REG_NUM;  
  
END
```

A-available
N-not available

BEFORE

<input type="checkbox"/>		Edit		Copy		Delete	ABX1234	ACTIVA	HONDA	2014	10000	Scooter	
<input type="checkbox"/>		Edit		Copy		Delete	ABX1234						

ACTIVATING TRIGGER (inserting in booking_details table)

+ Options		BOOKING_ID	FROM_DT_TIME	RET_DT_TIME	AMOUNT	BOOKING_STATUS	PICKUP_LOC	DROP_LOC	REG_NUM	DL_NUM	INS_CODE	ACT_RET_DT_TIM						
<input type="checkbox"/>		Edit		Copy		Delete	B1001	2022-01-20	2022-01-25	150.00		L101	L101		F1234554	NULL	2022-01-25 10:00:00	
<input type="checkbox"/>		Edit		Copy		Delete	B1002	2022-01-21	2022-01-24	90.00		L102	L102		SDF4567	T0981237	NULL	NULL

AFTER

<input type="checkbox"/>		Edit		Copy		Delete	ABX1234	ACTIVA	HONDA	2014	10000	Scooter	
<input type="checkbox"/>		Edit		Copy		Delete	ABX1234						

<input type="checkbox"/>		Edit		Copy		Delete	AHK7325	RAV4	BAJAJ	2022	3400	Cruiser Elite	
<input type="checkbox"/>		Edit		Copy		Delete	AHK7325						

2.UPDATE BIKE DETAILS

Book_bike trigger is used to set the availability_flag of a bike to “**available (A)**”, when booking is cancelled.

This trigger also updates pickup location of the bike to new pickup location when the bike is returned .

```
CREATE TRIGGER ` UPDATE_BIKE_DETAILS` AFTER UPDATE ON `booking_details`
FOR EACH ROW BEGIN
    IF NEW.BOOKING_STATUS = 'C' THEN
        UPDATE BIKE SET AVAILABILITY_FLAG = 'A' WHERE REGISTRATION_NUMBER =
NEW.REG_NUM;
    ELSE
        IF NEW.BOOKING_STATUS = 'R' THEN
            UPDATE BIKE SET LOC_ID = NEW.DROP_LOC ,AVAILABILITY_FLAG = 'A' WHERE
REGISTRATION_NUMBER = NEW.REG_NUM;
        END IF;
    END IF;
END
```

BEFORE

<input type="checkbox"/>				ABX1234	ACTIVA	HONDA	2014	10000	Scooter	N
<input type="checkbox"/>				AHK7325	RAV4	BAJAJ	2022	3400	Cruiser Elite	N
<input type="checkbox"/>				ASD9090	ACCORD	HONDA	2022	200	Cruiser	L103 N
<input type="checkbox"/>				CFT1908	328I	BMW	2015	10800	LUXURY BIKE	

ACTIVATING TRIGGER (cancelling booking)

+ Options	BOOKING_ID	FROM_DT_TIME	RET_DT_TIME	AMOUNT	BOOKING_STATUS	PICKUP_LOC	DROP_LOC	REG_NUM	DL_NUM	INS_CODE	ACT_RET_DT_TI
<input type="checkbox"/>	B1001	2022-01-20	2022-01-25	150.00	C	L101	L101	ABX1234	F1234554	NULL	2022-01-25 10:00:
<input type="checkbox"/>	B1002	2022-01-21	2022-01-24	90.00	C	L102	L102	SDF4567	T0981237	NULL	NULL

AFTER

<input type="checkbox"/> Edit Copy Delete ABX1234	ACTIVA	HONDA	2014	10000	Scooter	A
<input type="checkbox"/> Edit Copy Delete AHK7325	RAV4	BAJAJ	2022	3400	Cruiser Elite	N
<input type="checkbox"/> Edit Copy Delete ASD9090	ACCORD	HONDA	2022	200	Cruiser	L103 N
<input type="checkbox"/> Edit Copy Delete CFT1908	328I	BMW	2015	10800	LUXURY BIKE	N
<input type="checkbox"/> Edit Copy Delete CXZ2356	AVENGER	DODGE	2015	5000	Cruiser	A

CURSOR

FOR STORING BILL INFO IN BACKUP BILL TABLE

1) CREATING BACKUP FOR USER TABLE

```
CREATING BACKUP FOR USER TABLE
CREATE TABLE `billbackup` (
`BILL_ID` int(11) ,
`BILL_DATE` DATE ,
`TOTAL_AMOUNT` int(11)
)
```

2) DECLARING CURSOR USING PROCEDURE

```
DELIMITER $$

CREATE PROCEDURE c()
BEGIN
DECLARE done int(11);
DECLARE BILL_ID int(11);
DECLARE BILL_DATE DATE;
DECLARE TOTAL_AMOUNT int(11);

DECLARE cur CURSOR FOR SELECT * FROM BILLING_DETAILS;
DECLARE CONTINUE HANDLER FOR NOT FOUND SET done = 1;
OPEN cur;
```

```

label: LOOP
    FETCH cur INTO BILL_ID,BILL_DATE,TOTAL_AMOUNT;
    INSERT INTO billbackup VALUES(BILL_ID,BILL_DATE,TOTAL_AMOUNT);
    IF done = 1 THEN LEAVE label;
END IF;
END LOOP;
CLOSE cur;
END;$$
DELIMITER ;

```

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0265 seconds.)

```

CREATE PROCEDURE `c`() BEGIN DECLARE done int(11); DECLARE BILL_ID VARCHAR(11); DECLARE BILL_DATE DATE; DECLARE TOTAL_AMOUNT int(11); DECLARE BILL_STATUS int(11); DECLARE
BOOKING_ID int(11); DECLARE DISCOUNT_AMOUNT int(11); DECLARE Late_fee int(11); DECLARE Late CHAR; DECLARE cur CURSOR FOR SELECT * FROM BILLING_DETAILS; DECLARE CONTINUE HANDLER
FOR NOT FOUND SET done = 1; OPEN cur; label: LOOP FETCH cur INTO BILL_ID,BILL_DATE,BILL_STATUS,DISCOUNT_AMOUNT,TOTAL_AMOUNT,Late_fee,BOOKING_ID,Late; INSERT INTO billbackup
VALUES(BILL_ID,BILL_DATE,TOTAL_AMOUNT); IF done = 1 THEN LEAVE label; END IF; END LOOP; CLOSE cur; END;

```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

Calling cursor

✓ MySQL returned an empty result set (i.e. zero rows). (Query took 0.0132 seconds.)

```
call c();
```

[\[Edit inline\]](#) [\[Edit\]](#) [\[Create PHP code\]](#)

Bills stored in backup table

✓ Showing rows 0 - 2 (3 total, Query took 0.0011 seconds.)

```
SELECT * FROM `billbackup`
```

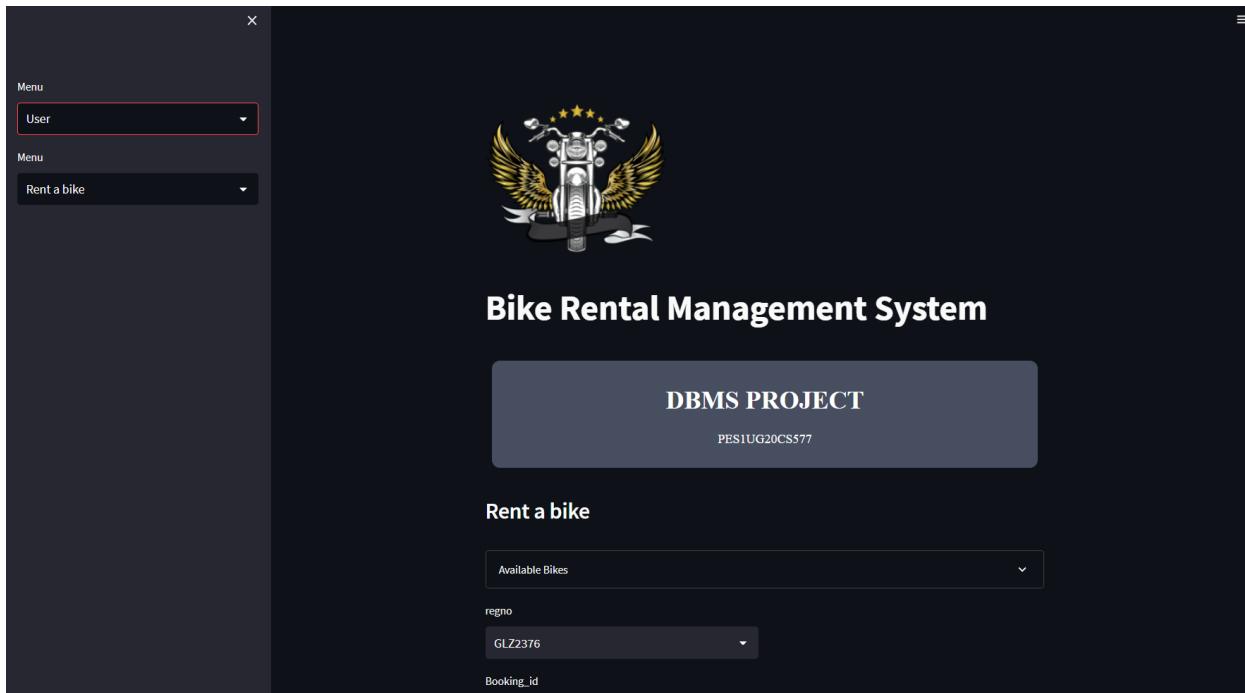
Profiling [\[Edit inline\]](#) [\[Edit\]](#) [\[Explain SQL\]](#) [\[Create PHP code\]](#) [\[Refresh\]](#)

Show all | Number of rows: 25 Search this table None

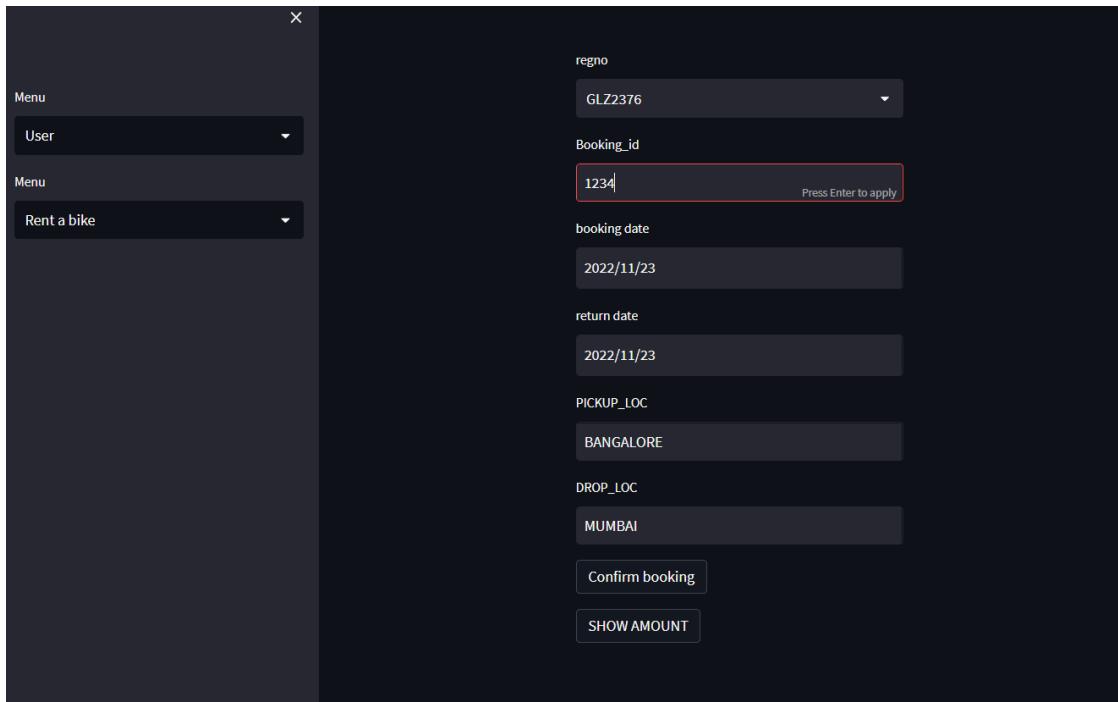
+ Options

		BILL_ID	BILL_DATE	TOTAL_AMOUNT					
<input type="checkbox"/>		Edit		Copy		Delete	BL1001	2022-01-25	138
<input type="checkbox"/>		Edit		Copy		Delete	BL1002	2022-01-15	487
<input type="checkbox"/>		Edit		Copy		Delete	RI 1003	2022-04-24	42

Frontend Screenshots



Booking a bike



CANCEL BOOKING

The screenshot shows a user interface for canceling a bike booking. On the left, there's a sidebar with a 'User' dropdown and a 'Cancel booking' option selected. The main area has a title 'Cancel booking' and a table titled 'booked bikes'. The table lists five rows of data:

	BOOKING_ID	FROM_DT_TIME	RET_DT_TIME	AMOUNT	BOOKING_STATUS	PICKUP_LOC	DROP_LOC
0	999	2022-11-23	2022-11-23	<NA>	B	<NA>	<NA>
1	B1005	2022-04-18	2022-04-25	26600	B	L102	L106
2	B1006	2022-04-21	2022-04-25	22800	B	L105	L107
3	B1007	2022-04-22	2022-04-25	40500	B	L102	L102
4	B1008	2022-04-11	2022-04-25	63000	B	L102	L102

Below the table, a dropdown menu shows 'regno' set to '999'. A red-bordered button labeled 'Cancel booking' is present. A green success message box says 'successfully canceled'. At the bottom, a smaller table shows the three remaining bookings:

	BOOKING_ID	FROM_DT_TIME	RET_DT_TIME
0	B1005	2022-04-18	2022-04-25
1	B1006	2022-04-21	2022-04-25
2	B1007	2022-04-22	2022-04-25

CRUD operation on some tables

bike table

The screenshot shows an 'Add Item' form for the 'Bike' table. The sidebar on the left has a 'Create' option selected. The main form has a title 'DBMS PROJECT' and 'PES1UG20CS577'. It contains several input fields:

- REGISTRATION_NUMBER: A numeric input field with value '0.00' and increment/decrement buttons.
- MODEL_NAME: A text input field.
- MAKE: A text input field.
- MODEL_YEAR: A numeric input field with value '0.00' and increment/decrement buttons.
- MILEAGE: A numeric input field with value '0.00' and increment/decrement buttons.
- BIKE_CATEGORY_NAME: A text input field.
- LOC_ID: A text input field.

CRUD operation on bike category table

The screenshot shows a dark-themed web application interface. At the top, there's a header bar with the text "DBMS PROJECT" and "PES1UG20CS577". On the left, a sidebar menu is visible with options: "Bike Category" (selected), "Create", "Read", "Update", and "Delete". The main content area has a title "Add Item". It contains several input fields with numerical values and increment/decrement buttons: "CATEGORY_NAME" (0.00), "BIKE_AGE" (empty), "LUGGAGE_CAPACITY" (empty), "COST_PER_DAY" (0.00), and "LATE_FEE_PER_HOUR" (0.00). A "submit" button is at the bottom.

CRUD operation on discount table

The screenshot shows a dark-themed web application interface. At the top, there's a header bar with the text "DBMS PROJECT" and "PES1UG20CS577". On the left, a sidebar menu is visible with options: "Discount" (selected), "Create", "Read", "Update" (highlighted with a red border), and "Delete". The main content area has a title "Edit Items". It contains a dropdown menu "Current Data", an input field "DISCOUNT_CODE" with value "D234", and two side-by-side input fields: "DISCOUNT_CODE" (D234) and "DISCOUNT_NAME" (CTS CORPORATE). Below these are "EXPIRY_DATE" (2023/09/14) and "DISCOUNT_PERCENTAGE" (20). A "Update Task" button is at the bottom, and a "View Updated Data" dropdown is also present.

SQL QUERRY BOX

The screenshot shows the homepage of the Bike Rental Management System. At the top center is a logo featuring a motorcycle engine with wings. Below the logo, the text "Bike Rental Management System" is displayed in bold white font. Underneath that, the text "bikerental query box" is also in bold white font. On the left side, there is a sidebar with two dropdown menus: "Menu" set to "sql command" and "Menu" set to "Home". The main content area has a dark background. On the left, there is a "HomePage" section. On the right, there is a "Query Submitted" section containing the SQL query "select * from bike;". Below this is a "Results" section showing a "Pretty Table" with the following data:

	0	1	2	3
24	QJ07621	EQUINOX	ROYALE NF	2013
25	QSC8709	MKZ	HAYABUSA	2012
26	QWE4562	LEGACY	HARLEY	2012

Thank you