**Task 1: Business Rules**

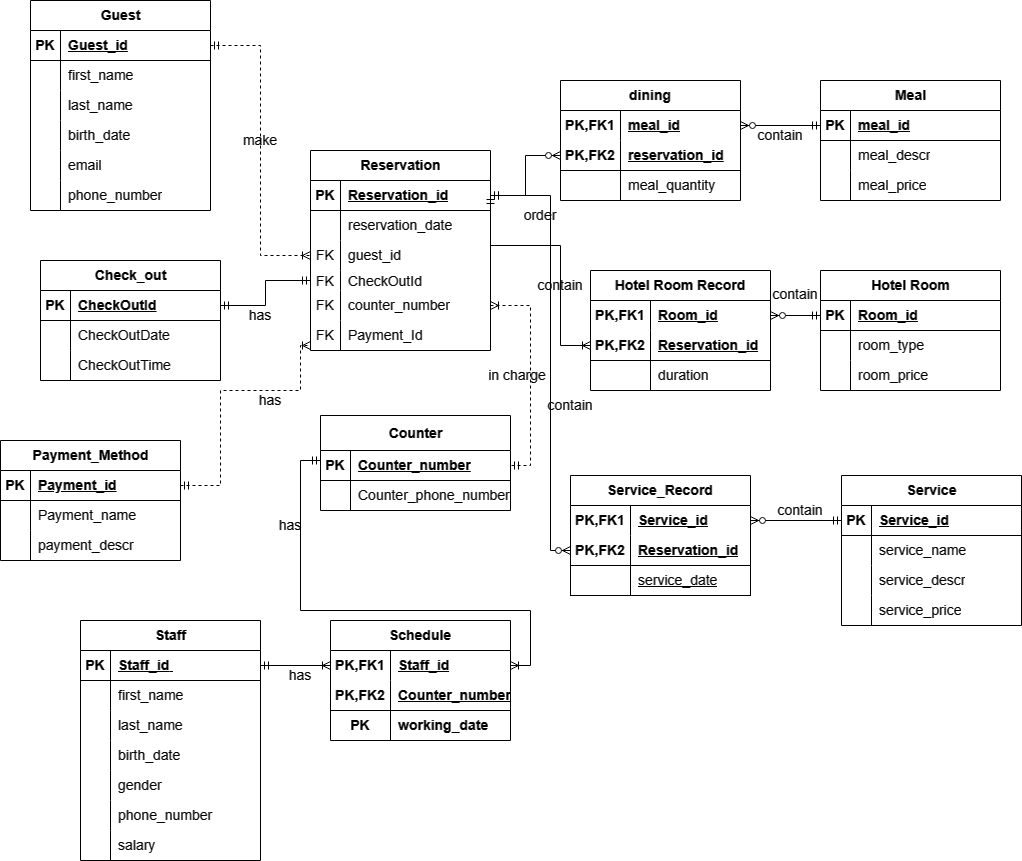
Business Rules

1. Counter is open from 7 am to 12 pm every day.
2. reservation date is equal to check in date
3. Customers must be registered as guests to use our service, and some customers registered as guests but didn’t use our service
4. Each reservation can order zero, one or many meals, and each meal can be ordered by one or many reservations.
5. Each reservation can use one or many service types, while one service type can offer one or many reservations
6. Each guest can have one or many reservations, while each reservation can only have one guest
7. Each reservation can have only one payment method, while each payment method can have one or many reservations
8. Each reservation will be handled at one counter, while each counter can handle many reservations
9. Each counter will be managed by one or many staff, while each staff will manage one or many counters
10. Each reservation can reserve one or many hotel rooms, while each hotel room can be reserved by many reservations
11. Each reservation has one check-out, while each check-out has one reservation

Assumption

1. Most guests will use delivery services when they decided to have meals
2. Every service is done by our staff

**Task 2: Entity Relationship Diagram**



**Task 3: the database schema**

|  |  |
| --- | --- |
| **Table** | **Data attribute** |
| Guest | guest(guest\_id, first\_name, last\_name, birth\_date, email, phone\_number) |
| Reservation | reservation(Reservation\_id, reservation\_date, CheckOutId \*, guest\_id\*, counter\_no\*, payment\_id\*) |
| Payment\_method | payment(Payment\_id, payment\_name, payment\_descr) |
| Meal | Meal(meal\_id, meal\_name, meal\_price) |
| Dining | Dining(meal\_id\*, Reservation\_id\*, meal\_qunatity) |
| Hotel Room Record | Hotel Room Record(Room\_id\*, Reservation\_id\*, duration) |
| Hotel Room | Hotel Room(Room\_id, room\_type, room\_price) |
| Service Record | service record(Service\_id\*, Reservation\_id\*, service\_date) |
| Service | service(Service\_id, service\_name, service\_descr, service\_price) |
| Check-out | CheckOut(CheckOutID, CheckOutDate, CheckOutTime) |
| Counter | counter (counter\_number, counter\_phone\_number) |
| Schedule | schedule (\*staff\_id, \*counter\_number, working\_date) |
| staff | Staff (staff\_id, first\_name, last\_name, gender, email, phone\_number, salary ) |

**Task 4 Create table**

4.1 Counter

create table counter (

counter\_number varchar(2) not null,

counter\_phone\_number varchar(17) not null,

primary key(counter\_number),

CONSTRAINT chk\_counter\_number CHECK (REGEXP\_LIKE(counter\_number,'^[A-Z]{1}\d{1}$'))

);

4.2 Staff

create table staff (

staff\_id varchar(4) not null,

first\_name varchar(12) not null,

last\_name varchar(12) not null,

email varchar(35) not null,

phone\_number varchar(17) not null,

salary number(7,2),

gender char(1) not null,

primary key (staff\_id),

constraint chk\_email check (REGEXP\_LIKE(email,'^[A-Za-z0-9\_%+-]+@[A-Za-z0-9-]+\.[A-Za-z]{2,}$')),

constraint chk\_gender check (UPPER(gender) in ('M','F'))

);

4.3 Schedule

create table schedule (

staff\_id varchar(4) not null,

counter\_number varchar(2) not null,

working\_day number(2) not null,

primary key(staff\_id,counter\_number,working\_day),

foreign key(counter\_number) references counter(counter\_number),

foreign key(staff\_id) references staff(staff\_id)

);

4.4 Check Out

CREATE TABLE CheckOut

(CheckOutID varchar(6) DEFAULT 0 CHECK(REGEXP\_LIKE(CheckOutId,'^[A-Z]{3}\d{3}$')) NOT NULL,

CheckOutDate DATE NOT NULL,

CheckOutTime varchar(15) DEFAULT "12:00 AM" NOT NULL,

PRIMARY KEY(CheckOutID));

4.5 Guest

CREATE TABLE Guest(

guest\_id number(4) NOT NULL

,first\_name VARCHAR(20) NOT NULL

,last\_name VARCHAR(20) NOT NULL

,birth\_date DATE NOT NULL

,guest\_email VARCHAR(50) NOT NULL

,phone\_number VARCHAR(13) NOT NULL,

primary key(guest\_id),

CONSTRAINT chk\_guest\_id CHECK (guest\_id >= 1001),

constraint chk\_guest\_email check (REGEXP\_LIKE(guest\_email,'^[A-Za-z0-9\_%+-]+@[A-Za-z0-9-]+\.[A-Za-z]{2,}$'))

);

4.6 Payment Method

create table Payment\_Method (

payment\_id number(2) not null,

payment\_name VARCHAR(25) NOT NULL,

payment\_descr VARCHAR(150) NOT NULL,

primary key(payment\_id),

CONSTRAINT chk\_payment\_id CHECK (payment\_id >= 1 AND payment\_id <=10)

);

4.7 reservation

create table Reservation (

reservation\_id VARCHAR(6) not null,

reservation\_date DATE not null,

payment\_id number(2) not null,

guest\_id number(4) not null,

counter\_number VARCHAR(2) not null,

CheckOutId VARCHAR(6) not null,

primary key(reservation\_id),

foreign key(payment\_id) references Payment\_Method(payment\_id),

foreign key(guest\_id) references Guest(guest\_id),

foreign key(counter\_number) references counter(counter\_number),

foreign key(CheckOutId) references CheckOut(CheckOutId),

CONSTRAINT chk\_reservation\_id CHECK (REGEXP\_LIKE(reservation\_id,'^[A-Z]{2}\d{4}$'))

);

4.8 service

create table service (

service\_id varchar(5) not null,

service\_name char(50) not null,

service\_descr varchar(150),

service\_price number(6,2) not null,

primary key(service\_id),

CONSTRAINT chk\_service\_id CHECK (REGEXP\_LIKE(service\_id,'^[A-Z]{1}\d{4}$'))

);

4.9 service record

create table service\_record (

service\_id varchar(5) not null,

reservation\_id varchar(6) not null,

service\_date date not null,

service\_time varchar(15) not null,

primary key(service\_id,reservation\_id, service\_date,service\_time),

foreign key(service\_id) references service(service\_id),

foreign key(reservation\_id) references reservation(reservation\_id)

);

4.10 meal

CREATE TABLE Meal

(meal\_id number(2) DEFAULT 1 CHECK(meal\_id >= 0) NOT NULL,

meal\_name varchar(20) default 'Meal' not null,

meal\_price NUMBER(5,2) DEFAULT 0.00 CHECK(meal\_price BETWEEN 0 AND 99.99) NOT NULL,

PRIMARY KEY(meal\_id));

4.11 Dining

CREATE TABLE Dining

(meal\_id number(2) DEFAULT 0 CHECK(meal\_id >= 0) NOT NULL,

Reservation\_id varchar(6) DEFAULT 0 CHECK(REGEXP\_LIKE(reservation\_id,'^[A-Z]{2}\d{4}$')) NOT NULL,

meal\_quantity number(2) default 0 NOT NULL,

PRIMARY KEY(meal\_id, Reservation\_id, meal\_quantity),

FOREIGN KEY(meal\_id) references Meal(meal\_id),

FOREIGN KEY(Reservation\_id) references reservation(reservation\_id));

4.12 Hotel Room

create table hotel\_room (

room\_id varchar(5) not null,

room\_type varchar(10) not null,

room\_price number(10,2) not null,

primary key(room\_id),

CONSTRAINT chk\_room\_id CHECK (REGEXP\_LIKE(room\_id,'^[A-Z]{1}\d{4}$'))

);

4.13 Hotel Room Record

create table hotel\_room\_record (

room\_id varchar(5) not null,

reservation\_id varchar(6) not null,

primary key(room\_id,reservation\_id),

foreign key(room\_id) references hotel\_room(room\_id),

foreign key(reservation\_id) references reservation(reservation\_id)

);

**Task 5 Insert data**

**5.1 Counter**

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('A1','+1 336 693 5156');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('A2','+62 446 184 9393');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('A3','+351 183 407 2865');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('A4','+233 505 942 1731');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('A5','+86 589 549 3213');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('B1','+58 323 712 8483');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('B2','+1 703 416 0387');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('B3','+7 521 267 6727');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('B4','+251 793 448 2253');

INSERT INTO counter(counter\_number,counter\_phone\_number) VALUES ('B5','+351 378 438 0650');

**5.2 Payment Method**

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (1, 'TNG eWallet', 'allow online and indoor');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (2, 'Master Card', 'allow online and indoor');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (3, 'Visa', 'allow online and indoor');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (4, 'PayPal', 'allow online');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (5, 'FPX', 'allow online');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (6, 'Alipay','allow online and indoor');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (7, 'Cash', 'allow indoor');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (8, 'American Express', 'allow online and indoor');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (9, 'telegraphic transfer', 'allow online');

insert into Payment\_Method (Payment\_id, payment\_name, payment\_descr) values (10, 'letter of credit', 'allow online and indoor');

**5.3 Check Out**

INSERT INTO CheckOut VALUES ('EVO349','28-May-2020','7:12:00 AM');

INSERT INTO CheckOut VALUES ('OVE290','21-Dec-2015','8:13:00 AM');

INSERT INTO CheckOut VALUES ('AWQ827','14-May-2019','11:27:00 AM');

INSERT INTO CheckOut VALUES ('HIV401','21-Mar-2019','8:03:00 AM');

INSERT INTO CheckOut VALUES ('DXB822','25-Nov-2019','9:28:00 AM');

INSERT INTO CheckOut VALUES ('CJI685','29-Apr-2017','11:58:00 AM');

INSERT INTO CheckOut VALUES ('YHR378','8-May-2023','7:26:00 AM');

INSERT INTO CheckOut VALUES ('AEB991','6-Mar-2020','7:35:00 AM');

INSERT INTO CheckOut VALUES ('LKE981','16-Apr-2018','8:40:00 AM');

INSERT INTO CheckOut VALUES ('PKH951','1-Jun-2020','7:50:00 AM');

INSERT INTO CheckOut VALUES ('SMU890','14-Sep-2016','10:45:00 AM');

**5.4 Guest**

INSERT INTO Guest VALUES (1001,'Sabrina','Beckingham','04-Mar-1977','sbeckingham0@myspace.com','+6014-3939995');

INSERT INTO Guest VALUES (1002,'Hermon','Dutch','03-Dec-1980','hdutch1@spiegel.de','+6012-3339866');

INSERT INTO Guest VALUES (1003,'Chiquita','Talton','29-Apr-1959','ctalton2@statcounter.com','+6014-1894379');

INSERT INTO Guest VALUES (1004,'Willem','Rossetti','22-Sep-1991','wrossetti3@archive.org','+6087-9589856');

INSERT INTO Guest VALUES (1005,'Doyle','Cutbush','16-Jun-1991','dcutbush4@biblegateway.com','+6080-8000499');

INSERT INTO Guest VALUES (1007,'Robby','Thickin','18-Dec-1983','rthickin6@blog.com','+6016-3447323');

INSERT INTO Guest VALUES (1008,'Keir','Antonucci','27-Sep-1957','kantonucci7@aol.com','+6016-6844685');

INSERT INTO Guest VALUES (1009,'Karil','Tullett','05-Oct-1964','ktullett8@t-online.de','+6014-7033659');

INSERT INTO Guest VALUES (1010,'Nial','Baskerville','24-Jun-2003','nbaskerville9@bandcamp.com','+603-5259686');

INSERT INTO Guest VALUES (1011,'Kelsey','De Domenico','01-Oct-1992','kdedomenicoa@bloglovin.com','+6013-9591235');

INSERT INTO Guest VALUES (1012,'Odell','Muckian','01-Apr-2003','omuckianb@dion.com','+6085-8933135');

**5.5 reservation**

INSERT INTO reservation VALUES ('UK7117','24-Apr-2020',8,1015,'A1','EVO349');

INSERT INTO reservation VALUES ('XB1308','10-Dec-2015',8,1017,'B1','OVE290');

INSERT INTO reservation VALUES ('MX9588','20-Mar-2019',9,1018,'B2','AWQ827');

INSERT INTO reservation VALUES ('HL8317','18-Mar-2019',1,1020,'B4','HIV401');

INSERT INTO reservation VALUES ('BF5792','17-Oct-2019',7,1011,'B5','DXB822');

INSERT INTO reservation VALUES ('BF6818','8-Mar-2017',9,1012,'B1','CJI685');

INSERT INTO reservation VALUES ('XI1297','27-Mar-2023',6,1010,'A2','YHR378');

INSERT INTO reservation VALUES ('HR4409','17-Feb-2020',7,1022,'A1','AEB991');

INSERT INTO reservation VALUES ('DX7910','12-Apr-2018',5,1011,'A5','LKE981');

INSERT INTO reservation VALUES ('KW1189','17-May-2020',1,1022,'A3','PKH951');

INSERT INTO reservation VALUES ('VV0429','17-Jul-2016',3,1012,'A3','SMU890');

**5.6 Staff**

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('D382','Cammy','Sweeting','csweeting0@qq.com','+86 103 299 0549',9094.4,'M');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('J758','Keeley','Sustins','ksustins1@google.ru','+55 632 716 4287',79128.5,'F');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('F476','Fenelia','Justham','fjustham2@paginegialle.it','+505 138 648 9003',32394.03,'F');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('K498','Virgil','Poulston','vpoulston3@prnewswire.com','+66 166 135 1577',96722.5,'M');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('G909','Avigdor','Mewitt','amewitt4@ameblo.jp','+55 370 898 8073',55914.2,'M');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('M726','Ernie','Bairnsfather','ebairnsfather5@chronoengine.com','+46 812 874 6001',34701.6,'M');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('I972','Alexander','Mourant','amourant6@wix.com','+351 589 465 9959',53264.71,'M');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('A272','Deny','Verrick','dverrick7@wisc.edu','+351 337 936 5382',19067.78,'F');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('J341','Kathe','McCrae','kmccrae8@huffingtonpost.com','+58 907 741 4802',8526.53,'F');

INSERT INTO staff(staff\_id,first\_name,last\_name,email,phone\_number,salary,gender) VALUES ('F513','Christel','Morais','cmorais9@hugedomains.com','+86 381 856 1535',43150.37,'F');

**5.7 service**

INSERT INTO service VALUES ('E6997','Front desk and concierge services','check-in and check-out',50);

INSERT INTO service VALUES ('L1214','Room service','order food and beverages',10);

INSERT INTO service VALUES ('V8040','Housekeeping',NULL,49.99);

INSERT INTO service VALUES ('W1729','Fitness centre',NULL,359.98);

INSERT INTO service VALUES ('V1322','Spa',NULL,429.99);

INSERT INTO service VALUES ('G5494','Business Centre','office equipment for guests to use',34.59);

INSERT INTO service VALUES ('G1852','Meeting and event spaces',NULL,899.88);

INSERT INTO service VALUES ('D6253','Laundry and dry-cleaning services',NULL,25.99);

INSERT INTO service VALUES ('F6126','In-room amenities',NULL,23.99);

INSERT INTO service VALUES ('B6201','Shuttle services',NULL,88.99);

**5.8 Meal**

INSERT INTO meal VALUES ('1','Garlic Bread',9.32);

INSERT INTO meal VALUES ('2','Noodle',79.82);

INSERT INTO meal VALUES ('3','Egg Fried Rice',21.87);

INSERT INTO meal VALUES ('4','Chicken Chop',18.76);

INSERT INTO meal VALUES ('5','Chicken Burger',19.27);

INSERT INTO meal VALUES ('6','Chicken Sandwitch',54.24);

INSERT INTO meal VALUES ('7','Salad',49.27);

INSERT INTO meal VALUES ('8','Muffin',16.43);

INSERT INTO meal VALUES ('9','Banana Ice cream',64.74);

INSERT INTO meal VALUES ('10','Hazelnut cake',37.22);

**5.9 Hotel Room**

INSERT INTO hotel\_room VALUES ('R0001','single',150);

INSERT INTO hotel\_room VALUES ('R0002','single',150);

INSERT INTO hotel\_room VALUES ('R0003','single',150);

INSERT INTO hotel\_room VALUES ('R0004','single',150);

INSERT INTO hotel\_room VALUES ('R0005','double',250);

INSERT INTO hotel\_room VALUES ('R0006','double',250);

INSERT INTO hotel\_room VALUES ('R0007','double',250);

INSERT INTO hotel\_room VALUES ('R0008','double',250);

INSERT INTO hotel\_room VALUES ('R0009','double',250);

INSERT INTO hotel\_room VALUES ('R0010','double',250);

INSERT INTO hotel\_room VALUES ('R0011','double',250);

**5.10 Hotel Room Record**

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0028','BH5341');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0007','QZ0211');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0008','VH4324');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0027','IM4245');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0014','YD7962');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0020','MA8181');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0028','HR4409');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0010','OX1748');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0007','LB7199');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0027','EX0916');

INSERT INTO hotel\_room\_record(room\_id,reservation\_id) VALUES ('R0012','XD4686');

**5.11 Schedule**

INSERT INTO schedule VALUES ('D382','A1',1);

INSERT INTO schedule VALUES ('J758','A2',1);

INSERT INTO schedule VALUES ('F476','A3',1);

INSERT INTO schedule VALUES ('K498','A4',1);

INSERT INTO schedule VALUES ('G909','A5',1);

INSERT INTO schedule VALUES ('M726','B1',1);

INSERT INTO schedule VALUES ('I972','B2',1);

INSERT INTO schedule VALUES ('A272','B3',1);

INSERT INTO schedule VALUES ('J341','B4',1);

INSERT INTO schedule VALUES ('F513','B5',1);

INSERT INTO schedule VALUES ('D382','A1',2);

**5.12 Service\_record**

insert into service\_record values ('W1729', 'FA3216', '19-Oct-2021', '4:18 AM');

insert into service\_record values ('G1852', 'BP6369', '22-Jul-2016', '1:05 AM');

insert into service\_record values ('V8040', 'PN6518', '15-Oct-2020', '12:04 AM');

insert into service\_record values ('G5494', 'BL3061', '19-Oct-2017', '4:23 AM');

insert into service\_record values ('F6126', 'ND6602', '3-Nov-2018', '12:04 AM');

insert into service\_record values ('V8040', 'GT8864', '13-Jun-2015', '3:59 AM');

insert into service\_record values ('V1322', 'RP7455', '13-Feb-2018', '6:31 AM');

insert into service\_record values ('V8040', 'JC2633', '16-Aug-2021', '12:24 AM');

insert into service\_record values ('G5494', 'BW1653', '5-Mar-2019', '3:33 AM');

insert into service\_record values ('V1322', 'VL3025', '3-Sep-2021', '4:39 AM');

insert into service\_record values ('G5494', 'BF5792', '17-Oct-2019', '3:34 AM');

**5.13 Dining**

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (5, 'DE1048', 2);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (2, 'XT9383', 10);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (10, 'GP8731', 9);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (3, 'VL3025', 7);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (9, 'GT8864', 6);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (8, 'GG8877', 8);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (3, 'QQ9992', 6);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (4, 'RK0542', 10);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (2, 'LV2501', 9);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (5, 'GW6365', 10);

insert into DIning (meal\_id, reservation\_id, meal\_quantity) values (5, 'RP7455', 4);

**Task 6 Query**

**6 .1**

**6.1.1 Query 1 : The total cost of one specific reservation**

Purpose : The purpose of this query is to calculate the total cost of one specific reservation. It can let an accountant to validate the cost is correct or not.

**SQL Code**

|  |
| --- |
| ACCEPT reservation\_id PROMPT 'Enter Reservation ID: '  column reservation\_id heading 'Reservation ID' format A14  column total\_service\_price heading 'Total Service Price(RM)' format 999,999.99  column total\_meal\_price heading 'Total Meal Price(RM)' format 999,999.99  column total\_room\_price heading 'Total Room Price(RM)' format 999,999.99  column total\_price heading 'Total Cost(RM)' format 999,999.99  TTITLE Center 'Total Cost For ' &reservation\_id -  Right 'Page: ' format 999 sql.pno skip 2  SELECT  E.reservation\_id,(  SELECT NVL(SUM(S.service\_price),0)  FROM service\_record R, service S  WHERE R.reservation\_id = E.reservation\_id  AND R.service\_id = S.service\_id  ) AS total\_service\_price,(  SELECT NVL(SUM(D.meal\_quantity \* M.meal\_price),0)  FROM dining D, meal M  WHERE D.reservation\_id = E.reservation\_id  AND D.meal\_id = M.meal\_id  ) AS total\_meal\_price,(  SELECT NVL(SUM(H.room\_price),0)  FROM hotel\_room\_record B, hotel\_room H  WHERE B.reservation\_id = E.reservation\_id  AND B.room\_id = H.room\_id  ) AS total\_room\_price,(  SELECT NVL(SUM(S.service\_price),0)  FROM service\_record R, service S  WHERE R.reservation\_id = E.reservation\_id  AND R.service\_id = S.service\_id  ) +(  SELECT NVL(SUM(D.meal\_quantity \* M.meal\_price),0)  FROM dining D, meal M  WHERE D.reservation\_id = E.reservation\_id  AND D.meal\_id = M.meal\_id  ) +(  SELECT NVL(SUM(H.room\_price),0)  FROM hotel\_room\_record B, hotel\_room H  WHERE B.reservation\_id = E.reservation\_id  AND B.room\_id = H.room\_id  ) AS total\_price  FROM reservation E  WHERE E.reservation\_id = '&reservation\_id'; |

Output

|  |
| --- |
| Total Cost For GT8864 Page: 1    Reservation ID Total Service Price(RM) Total Meal Price(RM) Total Room Price(RM) Total Cost(RM)  -------------- ----------------------- -------------------- -------------------- --------------  GT8864 49.99 1,325.40 1,150.00 2,525.39 |

**6.1.2 Query 2 : The total usage of one payment method**

Purpose : The purpose of this query is to indicate what is the most popular payment method used by customer. It can let the manager of the hotel to decide which kind of payment method must be focus in the future.

**SQL Code**

|  |
| --- |
| ACCEPT payment\_id PROMPT 'Enter Payment ID(1-10): '  column payment\_id heading 'Payment ID' format 99  column payment\_name heading 'Payment Name' format A30  column 'Full Name' format A30  column 'Subtotal Usage' format 999  TTITLE Center 'Total Number For Payment Method ' &payment\_id -  Right 'Page: ' format 999 sql.pno skip 2  Break On payment\_id on payment\_name skip 1  compute sum label 'Total: ' of "Subtotal Usage" on payment\_id  select R.payment\_id, payment\_name, First\_Name || ' ' || Last\_Name As "Full Name", count(R.payment\_id) as "Subtotal Usage"  from reservation R, guest G, payment\_method P  where R.guest\_id = G.guest\_id and R.payment\_id = P.payment\_id and R.payment\_id = &payment\_id  group by R.payment\_id, payment\_name, First\_Name || ' ' || Last\_Name  order by First\_Name || ' ' || Last\_Name; |

Output

|  |
| --- |
| Total Number For Payment Method 4 Page: 1    Payment ID Payment Name Full Name Subtotal Usage  ---------- ------------------------------ ------------------------------ --------------  4 PayPal Clyve Sherr 1  Grant Kobu 1  Lorne Bilney 1  Nial Baskerville 1  Odell Muckian 1  Sabrina Beckingham 1  Tabbitha Iddison 1    \*\*\*\*\*\*\*\*\*\* \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\* --------------  Total: 7 |

**6.1.3 Query 3 : Rank for total spending of guest**

Purpose : The purpose of this query is to find out customer spend how many amount in the hotel. The hotel will decide what kind of gift will send to them based on their spending in the hotel.

**SQL Code**

|  |
| --- |
| ACCEPT rownum PROMPT 'Enter the number of rank you want to display: '  column guest\_id heading 'Guest ID' format 9999  column 'Full Name' format A30  column total\_spending heading 'Total Spending(RM)' format 999,999.99  TTITLE Center 'Rank for total spending of guest Top ' &rownum-  Right 'Page: ' format 999 sql.pno skip 2  select \*  from(  select E.guest\_id, First\_Name || ' ' || Last\_Name as "Full Name" ,sum((  SELECT NVL(SUM(S.service\_price),0)  FROM service\_record R, service S  WHERE R.reservation\_id = E.reservation\_id  AND R.service\_id = S.service\_id  ) +(  SELECT NVL(SUM(D.meal\_quantity \* M.meal\_price),0)  FROM dining D, meal M  WHERE D.reservation\_id = E.reservation\_id  AND D.meal\_id = M.meal\_id  ) +(  SELECT NVL(SUM(H.room\_price),0)  FROM hotel\_room\_record B, hotel\_room H  WHERE B.reservation\_id = E.reservation\_id  AND B.room\_id = H.room\_id  )) AS total\_spending  from guest G, reservation E  where E.guest\_id = G.guest\_id  group by E.guest\_id, First\_Name || ' ' || Last\_Name  order by sum((  SELECT NVL(SUM(S.service\_price),0)  FROM service\_record R, service S  WHERE R.reservation\_id = E.reservation\_id  AND R.service\_id = S.service\_id  ) +(  SELECT NVL(SUM(D.meal\_quantity \* M.meal\_price),0)  FROM dining D, meal M  WHERE D.reservation\_id = E.reservation\_id  AND D.meal\_id = M.meal\_id  ) +(  SELECT NVL(SUM(H.room\_price),0)  FROM hotel\_room\_record B, hotel\_room H  WHERE B.reservation\_id = E.reservation\_id  AND B.room\_id = H.room\_id  )) desc  )  where rownum <= &rownum; |

Output

|  |
| --- |
| Rank for total spending of guest Top 5 Page: 1    Guest ID Full Name Total Spending(RM)  -------- ------------------------------ ------------------  1003 Chiquita Talton 20,369.28  1014 Lorne Bilney 17,868.85  1010 Nial Baskerville 13,494.34  1022 Geoffrey Gotcher 12,461.00  1002 Hermon Dutch 12,304.05 |

**6.2.1 Query 1 : Show the services which have a total price exceeding the specific number.**

Purpose: The purpose of this report is to check all the service records which exceed the number inputted by employees for employees understanding which services customers prefer to spend more money on.

**SQL Code**

|  |
| --- |
| ACCEPT specific\_price PROMPT 'Total Price Exceeding: '  COLUMN service\_id heading 'Service ID' FORMAT A10  COLUMN service\_name heading 'Service Name' FORMAT A50  COLUMN SUM(service\_price) heading 'Total Service Price(RM)' FORMAT 999,999.99  TTITLE LEFT 'THE SERVICES WHICH HAVE A TOTAL PRICE EXCEEDING ' &specific\_price -  RIGHT 'Page: ' FORMAT 999 sql.pno skip 2  SET PAGESIZE 15  SET LINESIZE 100  SELECT S.service\_id, service\_name, SUM(service\_price)  FROM service S, service\_record R  WHERE S.service\_id = R.service\_id  GROUP BY S.service\_id, service\_name  HAVING SUM(service\_price) > &specific\_price  ORDER BY SUM(service\_price); |

**Output**

|  |
| --- |
| THE SERVICES WHICH HAVE A TOTAL PRICE EXCEEDING 500 Page: 1  Service ID Service Name Total Service Price(RM)  ---------- -------------------------------------------------- -----------------------  F6126 In-room amenities 599.75  D6253 Laundry and dry-cleaning services 935.64  G5494 Business Centre 1,176.06  V8040 Housekeeping 1,649.67  E6997 Front desk and concierge services 1,850.00  B6201 Shuttle services 2,224.75  W1729 Fitness centre 12,959.28  V1322 Spa 13,759.68  G1852 Meeting and event spaces 17,997.60 |

**6.2.2 Query 2 : Show the services which are more than a specific number of services at all times.**

Purpose: The purpose of this report is to check all the service records which are more than the number inputted by employees for employees to find out which service the customer is more interested in.

**SQL Code**

|  |
| --- |
| ACCEPT service\_count PROMPT 'Enter the number of service (0-40): '  COLUMN service\_id heading 'Service ID' FORMAT A10  COLUMN service\_name heading 'Service Name' FORMAT A50  COLUMN service\_count heading 'Number of Services' FORMAT 99  TTITLE LEFT 'NUMBER OF SERVICE MORE THAN ' &service\_count ' TIMES AT ALL TIMES '  RIGHT 'Page: ' FORMAT 999 sql.pno skip 2  SET PAGESIZE 15  SET LINESIZE 100  SELECT S.service\_id, service\_name, COUNT(S.service\_id) as service\_count  FROM service S, service\_record R  WHERE S.service\_id = R.service\_id  GROUP BY S.service\_id, service\_name  HAVING COUNT(S.service\_id) > &service\_count  ORDER BY COUNT(S.service\_id) DESC; |

**Output**

|  |
| --- |
| NUMBER OF SERVICE MORE THAN 25 TIMES AT ALL TIMES  Service ID Service Name Number of Services  ---------- -------------------------------------------------- ------------------  E6997 Front desk and concierge services 37  W1729 Fitness centre 36  D6253 Laundry and dry-cleaning services 36  G5494 Business Centre 34  V8040 Housekeeping 33  V1322 Spa |

**6.2.3 Query 3 : Show the details(id, name, date, time) of the specific service id.**

Purpose: The purpose of this report is to check the details of the specific service id which is inputted by the employees.

**SQL Code**

|  |
| --- |
| ACCEPT service\_id PROMPT 'Enter Service ID: '  COLUMN service\_id heading 'Service ID' FORMAT A10  COLUMN service\_name heading 'Service Name' FORMAT A50  COLUMN service\_date heading 'Service Date' FORMAT A12  COLUMN service\_time heading 'Service Time' FORMAT A12  TTITLE LEFT 'SERVICE DETAILS OF SERVICE ID ' &service\_id -  RIGHT 'Page: ' FORMAT 999 sql.pno skip 2  SET PAGESIZE 15  SET LINESIZE 100  SELECT S.service\_id, service\_name, service\_date, service\_time  FROM service S, service\_record R  WHERE R.service\_id = S.service\_id AND R.service\_id LIKE '&service\_id'; |

**Output**

|  |
| --- |
| SERVICE DETAILS OF SERVICE ID E6997 Page: 1  Service ID Service Name Service Date Service Time  ---------- -------------------------------------------------- ------------ ------------  E6997 Front desk and concierge services 09-NOV-17 12:59 AM  E6997 Front desk and concierge services 17-OCT-19 5:06 AM  E6997 Front desk and concierge services 08-MAR-17 6:40 AM  E6997 Front desk and concierge services 19-OCT-17 12:33 AM  E6997 Front desk and concierge services 22-JUL-16 4:13 AM  E6997 Front desk and concierge services 05-MAR-19 5:42 AM  E6997 Front desk and concierge services 08-OCT-21 6:53 AM  E6997 Front desk and concierge services 12-APR-18 3:06 AM  E6997 Front desk and concierge services 18-MAR-19 6:54 AM  E6997 Front desk and concierge services 01-FEB-17 6:08 AM  SERVICE DETAILS OF SERVICE ID E6997 Page: 2  Service ID Service Name Service Date Service Time  ---------- -------------------------------------------------- ------------ ------------  E6997 Front desk and concierge services 27-APR-20 5:54 AM  E6997 Front desk and concierge services 18-MAR-19 5:28 AM  E6997 Front desk and concierge services 28-APR-22 3:16 AM  E6997 Front desk and concierge services 11-FEB-21 12:15 AM  E6997 Front desk and concierge services 17-MAY-20 12:13 AM  E6997 Front desk and concierge services 17-MAY-20 5:01 AM  E6997 Front desk and concierge services 09-OCT-18 6:41 AM  E6997 Front desk and concierge services 03-NOV-18 12:37 AM  E6997 Front desk and concierge services 15-OCT-20 5:48 AM  E6997 Front desk and concierge services 05-JUL-20 4:59 AM  SERVICE DETAILS OF SERVICE ID E6997 Page: 3  Service ID Service Name Service Date Service Time  ---------- -------------------------------------------------- ------------ ------------  E6997 Front desk and concierge services 11-MAR-18 2:35 AM  E6997 Front desk and concierge services 11-MAR-18 2:43 AM  E6997 Front desk and concierge services 19-MAY-20 5:28 AM  E6997 Front desk and concierge services 30-SEP-17 1:14 AM  E6997 Front desk and concierge services 03-OCT-21 4:05 AM  E6997 Front desk and concierge services 03-OCT-21 6:10 AM  E6997 Front desk and concierge services 21-APR-20 5:20 AM  E6997 Front desk and concierge services 10-AUG-15 6:40 AM  E6997 Front desk and concierge services 03-SEP-21 2:00 AM  E6997 Front desk and concierge services 03-SEP-21 3:57 AM  SERVICE DETAILS OF SERVICE ID E6997 Page: 4  Service ID Service Name Service Date Service Time  ---------- -------------------------------------------------- ------------ ------------  E6997 Front desk and concierge services 24-JUL-18 4:04 AM  E6997 Front desk and concierge services 10-DEC-15 1:45 AM  E6997 Front desk and concierge services 02-NOV-15 12:23 AM  E6997 Front desk and concierge services 27-MAR-23 3:36 AM  E6997 Front desk and concierge services 09-OCT-17 6:08 AM  E6997 Front desk and concierge services 28-JAN-17 12:35 AM  E6997 Front desk and concierge services 28-JAN-17 6:57 AM |

**6 .3**

1. get all reservation and payment name of meal with input meal ID

to understand how customer pay eg use credit card or cash, so can entertain customer refund using same method

SQL

|  |
| --- |
| ACCEPT ENUMBER NUMBER PROMPT 'Meal ID. :'  SET LINESIZE 120  SET PAGESIZE 30  COLUMN payment\_name FORMAT A25  COLUMN payment\_descr FORMAT A50 TRUNCATE  TTITLE CENTER 'Customer Payment Method'  RIGHT 'Page:' FORMAT 999 SQL.PNO SKIP 2  BREAK ON payment\_id SKIP 1  COMPUTE COUNT OF payment\_name ON payment\_descr  select R.reservation\_id, R.payment\_id, payment\_name, payment\_descr  from Payment\_method P, Reservation R, Dining D, Meal M  where M.meal\_id=&ENUMBER and P.Payment\_id = R.Payment\_id and R.Reservation\_id = D.Reservation\_id and D.meal\_id = M.meal\_id  order by payment\_name; |

Output

|  |
| --- |
| Customer Payment Method  Reservation ID Payment ID Payment Name PAYMENT\_DESCR  ------------------------------ ---------- ------------------------- --------------------------------------------------  XI1297 6 Alipay allow online and indoor  ER4237 8 American Express allow online and indoor  YN0014 American Express allow online and indoor  LD3924 American Express allow online and indoor  ER4237 American Express allow online and indoor  XB1308 American Express allow online and indoor  YD7962 7 Cash allow indoor  YD7962 Cash allow indoor  XD4686 5 FPX allow online  OG2085 FPX allow online  PN6518 FPX allow online |

2. get number of guests who ordered input meal id

to understand how popular the meal is so can adjust meal stock eg increase stock if high demand to avoid sold out and increase profit

SQL

|  |
| --- |
| ACCEPT ENUMBER NUMBER PROMPT 'Meal ID. :'  SET LINESIZE 120  SET PAGESIZE 30  COLUMN meal\_id FORMAT 99  COLUMN TotalNumOfOrders FORMAT 999 TRUNCATE  TTITLE CENTER 'Meal Popularity'  RIGHT 'Page:' FORMAT 999 SQL.PNO SKIP 2  BREAK ON REPORT  COMPUTE SUM LABEL TOTAL OF TotalNumOfOrders ON REPORT  SELECT D.meal\_id, COUNT(R.guest\_id) AS TotalNumOfOrders  FROM Guest G, reservation R, Dining D, Meal M  WHERE G.guest\_id=R.guest\_id and R.reservation\_id = D.reservation\_id and D.meal\_id = M.meal\_id and M.meal\_id=&ENUMBER  group by D.meal\_id  order by D.meal\_id； |

Output

|  |
| --- |
| Meal Popularity  MEAL\_ID TOTALNUMOFORDERS  ------- ----------------  1 26  ----------------  TOTAL 26 |

3. get check out date of input reservation id

to prepare check out documents for customer and room service to immediately clean room after check out so that next customer can immediately check in and use room.

SQL

|  |
| --- |
| ACCEPT ENUMBER PROMPT 'Reservation Id. :'  SET LINESIZE 120  SET PAGESIZE 30  COLUMN CheckOutDate FORMAT A12  COLUMN CheckOutID FORMAT A10  TTITLE CENTER 'Meal Popularity'  RIGHT 'Page:' FORMAT 999 SQL.PNO SKIP 2  BREAK ON REPORT  SELECT R.reservation\_id, R.CheckOutID, CheckOutDate, R.guest\_id, R.counter\_number  FROM CheckOut C, Guest G, reservation R  WHERE G.guest\_id=R.guest\_id and R.CheckOutID = C.CheckOutID and R.reservation\_id='&ENUMBER'  order by CheckOutID; |

Output

|  |
| --- |
| Meal Popularity  Reservation ID Check Out Check Out Da Guest ID Counter Number  ------------------------------ ---------- ------------ -------- ---------------  UK7117 EVO349 28-MAY-20 1015 A1 |

**6.4**

**6.4.1 Query 1 : The quantity of guest for each service**

Purpose : The purpose of this query is find out the quantity of guest for each service, manager know how many quantity of guest, easy to know which service is more popular and easy to know it should hire how many staff in future when there is too much customers.

**SQL Code**

|  |
| --- |
| column C1 heading "Service Name" format a35;  column C2 heading "Service ID " format a10;  TTITLE CENTER 'Selected customer quantity of service name' skip 2  ACCEPT service\_name PROMPT 'Enter sevice\_name : '  SELECT count(G.guest\_id) as QuantityofCustomer, service\_name C1, S.service\_id C2  FROM guest G , reservation R , service\_record S , service T  WHERE G.guest\_id = R.guest\_id AND R.reservation\_id = S.reservation\_id AND S.service\_id = T.service\_id AND service\_name = '&service\_name'  GROUP BY service\_name , S.service\_id; |

Output

|  |
| --- |
| Selected customer quantity of service name  QUANTITYOFCUSTOMER Service Name Service ID  ------------------ ----------------------------------- ----------  32 Spa V1322 |

**6.4.2 Query 2 : The staff who make specified reservation**

Purpose : The purpose of this query is find out the staff who make the reservation for customers.When there is facing some problem or staff make a wrong reservation for customer, manager able to check back easily.

**SQL Code**

|  |
| --- |
| column C1 heading "STAFF ID " format a10 ;  column C2 heading "First Name " format a10 ;  column C3 heading "Last Name" format a10  column C4 heading "Counter Number" format a14;  column C5 heading "Reservation ID" format a16 ;  TTITLE center ' Selected Staff ID ' skip 2  ACCEPT reservation\_id prompt "Enter reservation ID: "  SELECT S.staff\_id C1, first\_name C2, last\_name C3, C.counter\_number C4, R.reservation\_id C5  FROM staff S , schedule D , counter C , reservation R  WHERE S.staff\_id = D.staff\_id AND D.counter\_number = C.counter\_number AND C.counter\_number = R.counter\_number AND reservation\_id = '&reservation\_id'  GROUP BY S.staff\_id , first\_name , last\_name , C.counter\_number , R.reservation\_id ; |

Output

|  |
| --- |
| Selected Staff ID  STAFF ID First Name Last Name Counter Number Reservation ID  ---------- ---------- ---------- -------------- ----------------  F513 Christel Morais B5 LV3148 |

**6.4.3 Query 3 : The top three popular meals .**

Purpose : The purpose of this query is to find out the top three popular meals. The chef at the kitchen able according to this result, to estimate how much raw material should be prepared. Manager also able to know which meals are more popular to attract customers.

**SQL Code**

|  |
| --- |
| column reservation heading "Reservation ID " format 999 ;  column meal\_id heading "Meal ID " format 999 ;  column meal\_name heading "Meal Name" format a30;  TTITLE CENTER 'The rank of Popular Meal' skip 2  ACCEPT ROWNUM PROMPT "Enter the rank of popular meal :"  SELECT \*  FROM (Select count(reservation\_id) as reservation , m.meal\_id, meal\_name  FROM dining d, meal m  WHERE d.meal\_id = m.meal\_id  GROUP BY m.meal\_id , meal\_name  ORDER BY count(reservation\_id) DESC )  WHERE ROWNUM <= &ROWNUM; |

Output

|  |
| --- |
| The rank of Popular Meal  Reservation ID Meal ID Meal Name  --------------- -------- ------------------------------  37 10 Hazelnut cake  35 8 Muffin  34 4 Chicken Chop |

6.5.1 The rank of popular room type

Purpose : Allow the manager to know which kind of room type is more popular for the future expansion of hotel.

SQL

|  |
| --- |
| ACCEPT rownum PROMPT 'Enter the rank of popular room type: '  column room\_id heading 'Room ID' format a10  column room\_type heading 'Room Type' format a20  column reservation\_amount heading 'Reservation Amount' 999  TTITLE Center 'Top Three Used Room Types'  SELECT \*  FROM (  SELECT count(reservation\_id) as reservation, r.room\_id, room\_type  FROM hotel\_room h, hotel\_room\_record r  WHERE h.room\_id = r.room\_id  GROUP BY r.room\_id, room\_type  ORDER BY count(reservation\_id) DESC)  where rownum <= &rownum; |

Output

|  |
| --- |
| Top Three Used Room Types  RESERVATION Room ID Room Type  ----------- ---------- --------------------  16 R0020 quad  15 R0018 triple  15 R0015 triple |

6.5.2 The total income of each room

Purpose : To inform the manager of hotel that which room is more popular. This can make the hotel to build more similar room in future expantion

SQL

|  |
| --- |
| column room\_id heading 'Room ID' format a10  column room\_type heading 'Room Type' format a10  column total\_income heading 'Total income' format 999,999.99  TTITLE Center 'Total income of each room'  select r.room\_id, room\_type ,sum(room\_price) As total\_room\_price  FROM hotel\_room h, hotel\_room\_record r  WHERE h.room\_id = r.room\_id  GROUP BY r.room\_id, room\_type; |

Output

|  |
| --- |
| Total income of each room Page: 1  Room ID Room Type Total Room Price(RM)  ---------- ---------- --------------------  R0002 single 1,350.00  R0023 quad 3,600.00  R0027 queen 6,000.00  R0006 double 2,000.00  R0014 double 3,000.00  R0018 triple 5,100.00  R0012 double 3,250.00  R0025 queen 7,200.00  R0003 single 1,950.00  R0005 double 3,250.00  R0007 double 2,750.00  R0008 double 3,500.00  R0013 double 1,750.00  R0029 king 4,000.00  R0016 triple 2,040.00  R0026 queen 6,000.00  R0028 king 10,400.00 |

6.5.3 The room reserve in each reservation

Purpose : To make validation of the key receive from the guest is similar to the to room the guest has reserved

SQL

|  |
| --- |
| ACCEPT reservation\_id PROMPT 'Enter Reservation ID: '  column reservation\_id heading 'Reservation ID' format a20  column room\_id heading 'Room ID' format a20  column room\_type heading 'Room Type' format a10  TTITLE Center 'The room reserve in reservation' &reservation\_id  break on reservation\_id  SELECT e.reservation\_id, r.room\_id, room\_type  FROM hotel\_room h, hotel\_room\_record r, reservation e  WHERE h.room\_id = r.room\_id and r.reservation\_id = e.reservation\_id AND e.reservation\_id = '&reservation\_id'; |

Output

|  |
| --- |
| The room reserve in reservationUK7117  Reservation ID Room ID Room Type  -------------------- -------------------- ----------  UK7117 R0026 queen |

**Task 7 Extra Work**

7.1.1 View for reservation table with total cost

Purpose: To allow easy view of the total cost of every reservation

SQL

|  |
| --- |
| column reservation\_id heading 'Reservation ID' format A14  column reservation\_date heading 'Reservation Date' format A15  column payment\_method heading 'Payment Method' format 99  column guest\_id heading 'Guest ID' format 9999  column counter\_number heading 'Counter Number' format A15  column CheckOutId heading 'Check Out ID' format A15  column total\_cost heading 'Total Cost(RM)' format 999,999.99  TTITLE Center 'Reservation table with total cost' -  Right 'Page: ' format 999 sql.pno skip 2  create or replace view reservation\_with\_price as  select E.reservation\_id, reservation\_date, E.payment\_id, E.guest\_id, E.counter\_number, E.CheckOutId,(  SELECT NVL(SUM(S.service\_price),0)  FROM service\_record R, service S  WHERE R.reservation\_id = E.reservation\_id  AND R.service\_id = S.service\_id  ) +(  SELECT NVL(SUM(D.meal\_quantity \* M.meal\_price),0)  FROM dining D, meal M  WHERE D.reservation\_id = E.reservation\_id  AND D.meal\_id = M.meal\_id  ) +(  SELECT NVL(SUM(H.room\_price),0)  FROM hotel\_room\_record B, hotel\_room H  WHERE B.reservation\_id = E.reservation\_id  AND B.room\_id = H.room\_id  ) AS total\_cost  FROM reservation E; |

Output

|  |
| --- |
| Reservation table with total cost Page: 1    Reservation ID Reservation Dat Payment ID Guest ID Counter Number Check Out ID Total Cost(RM)  -------------- --------------- ---------- -------- --------------- --------------- --------------  UK7117 24-APR-20 8 1015 A1 EVO349 1,450.34  XB1308 10-DEC-15 8 1017 B1 OVE290 1,648.17  MX9588 20-MAR-19 9 1018 B2 AWQ827 2,226.75  HL8317 18-MAR-19 1 1020 B4 HIV401 1,495.94  BF5792 17-OCT-19 7 1011 B5 DXB822 1,712.88  BF6818 08-MAR-17 9 1012 B1 CJI685 1,440.00  XI1297 27-MAR-23 6 1010 A2 YHR378 1,838.81  HR4409 17-FEB-20 7 1022 A1 AEB991 2,228.29  DX7910 12-APR-18 5 1011 A5 LKE981 3,103.52  KW1189 17-MAY-20 1 1022 A3 PKH951 2,038.13 |

7.1.2 View for the history of each guest use what method to pay

Purpose: To know what payment method have been used by each guest at their past reservation

SQL

|  |
| --- |
| column 'Full Name' format A30  column reservation\_id heading 'Reservation ID' format A30  column payment\_name heading 'Payment Name' format A30  column CheckOutDate heading 'Check Out Date' format A15  TTITLE Center 'The history of each guest using what kind of method to pay' -  Right 'Page: ' format 999 sql.pno skip 2  Break On 'Full Name' skip 1  create or replace view history\_of\_payment\_method as  select First\_Name || ' ' || Last\_Name as "Full Name", R.reservation\_id, CheckOutDate, payment\_name  from guest G, CheckOut C, reservation R, Payment\_method P  where G.guest\_id = R.guest\_id and C.checkOutId = R.checkOutId and R.payment\_id = P.payment\_id  order by First\_Name || ' ' || Last\_Name, CheckOutDate; |

Output

|  |
| --- |
| The history of each guest using what kind of method to pay Page: 1    Full Name Reservation ID Check Out Date Payment Name  ------------------------------ ------------------------------ --------------- ------------------------------  Alfie Golightly XB1308 21-DEC-15 American Express  GG8877 15-OCT-16 letter of credit  YD7962 08-MAY-18 Cash  SE5736 17-SEP-21 telegraphic transfer    Allyson Concannon ST0707 08-JUN-15 Visa  VJ9868 31-OCT-16 Master Card  RT2938 22-NOV-17 American Express  EX0916 06-FEB-19 Alipay  GW8876 23-AUG-19 Visa    Biron Le land ZJ4688 09-JUN-15 FPX |

7.2.1 view how many reservation of room in each year

Purpose: Manger easy to take record from here for making any report to he boss

SQL

|  |
| --- |
| column 'id' heading 'Number Reservation ID' format 99  column 'year' heading 'YEAR' format 9999  column 'room\_id' heading 'Room ID' format A20  TTITLE Center 'Quantity of room reserved in Year' skip 2  create or replace view quantity\_of\_room as  select count (R.reservation\_id) as id , extract (year from reservation\_date) as year , room\_id  from reservation R , hotel\_room\_record HR  where R.reservation\_id = HR.reservation\_id  group by extract (year from reservation\_date), room\_id  order by year ; |

Output

|  |
| --- |
| Quantity of room reserved in Year  Number Reservation ID YEAR Room ID  --------------------- ----- --------------------  1 2015 R0003  3 2015 R0004  2 2015 R0005  1 2015 R0006  1 2015 R0007  1 2015 R0008  1 2015 R0009  4 2015 R0010  3 2015 R0011  3 2015 R0012  1 2015 R0013  1 2015 R0014  1 2015 R0015  2 2015 R0018  1 2015 R0019  1 2015 R0020  1 2015 R0021  1 2015 R0023  1 2015 R0024  2 2015 R0025  2 2015 R0026  2 2015 R0027  1 2015 R0028  1 2016 R0001  1 2016 R0004  1 2016 R0006  3 2016 R0007  2 2016 R0008  3 2016 R0010  2 2016 R0011  1 2016 R0012  1 2016 R0016  1 2016 R0017  1 2016 R0020  1 2016 R0021  1 2016 R0023  2 2016 R0024  3 2016 R0025  1 2016 R0026  1 2017 R0002  1 2017 R0003  1 2017 R0004  1 2017 R0005  1 2017 R0007  2 2017 R0008  1 2017 R0009  1 2017 R0010  1 2017 R0011  3 2017 R0012  1 2017 R0014  2 2017 R0015  1 2017 R0017  3 2017 R0018  1 2017 R0019  1 2017 R0020  2 2017 R0021  1 2017 R0024  1 2017 R0027  2 2017 R0028  1 2017 R0029  1 2017 R0030  1 2018 R0001  1 2018 R0002  2 2018 R0003  3 2018 R0004  2 2018 R0005  2 2018 R0007  1 2018 R0008  2 2018 R0009  1 2018 R0010  2 2018 R0011  1 2018 R0012  1 2018 R0014  1 2018 R0015  1 2018 R0016  1 2018 R0017  2 2018 R0018  3 2018 R0019  4 2018 R0020  2 2018 R0021  2 2018 R0023  4 2018 R0024  2 2018 R0025  2 2018 R0026  2 2018 R0027  2 2018 R0028  1 2018 R0030  1 2019 R0001  2 2019 R0002  5 2019 R0003  1 2019 R0004  1 2019 R0005  3 2019 R0006  1 2019 R0007  2 2019 R0008  2 2019 R0009  2 2019 R0012  2 2019 R0013  3 2019 R0014  6 2019 R0015  1 2019 R0017  2 2019 R0018  1 2019 R0019  3 2019 R0020  2 2019 R0023  1 2019 R0024  2 2019 R0026  2 2019 R0027  1 2019 R0028  1 2019 R0029  1 2019 R0030  1 2020 R0001  1 2020 R0003  2 2020 R0005  1 2020 R0007  1 2020 R0008  1 2020 R0012  1 2020 R0013  1 2020 R0014  3 2020 R0015  2 2020 R0016  1 2020 R0018  5 2020 R0019  1 2020 R0020  2 2020 R0021  1 2020 R0024  3 2020 R0025  3 2020 R0026  3 2020 R0028  2 2020 R0030  2 2021 R0001  4 2021 R0002  1 2021 R0003  2 2021 R0005  2 2021 R0006  3 2021 R0008  2 2021 R0010  2 2021 R0011  3 2021 R0014  1 2021 R0015  2 2021 R0016  2 2021 R0017  1 2021 R0018  2 2021 R0019  2 2021 R0020  1 2021 R0021  1 2021 R0022  1 2021 R0023  1 2021 R0025  1 2021 R0027  2 2021 R0028  2 2021 R0029  2 2021 R0030  1 2022 R0003  1 2022 R0004  1 2022 R0005  1 2022 R0006  2 2022 R0007  2 2022 R0008  1 2022 R0009  1 2022 R0010  1 2022 R0011  2 2022 R0012  2 2022 R0013  2 2022 R0014  1 2022 R0015  1 2022 R0017  3 2022 R0018  1 2022 R0019  2 2022 R0020  1 2022 R0023  2 2022 R0027  1 2022 R0028  1 2022 R0029  1 2023 R0001  1 2023 R0002  1 2023 R0005  1 2023 R0013  1 2023 R0028 |

7.2.2 View to category of the room price that have been reserved

Purpose: Manger able to distinguish each category or how many high price room are being reserved

SQL

|  |
| --- |
| column 'roomprice' format A5  column room\_type format A9  column reservation\_id heading 'Reservation ID' format A20  TTITLE Center 'The price of Reserved ROOM' skip 2  create or replace view price\_of\_reservedroom as  select reservation\_id, room\_type ,concat('$' ,room\_price) AS roomprice ,  case  when room\_price > 600 THEN 'High'  WHEN room\_price > 400 THEN 'Medium'  ELSE 'Low'  END AS Price  FROM hotel\_room H, hotel\_room\_record HR  WHERE H.room\_id = HR.room\_id ; |

Output

|  |
| --- |
| The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  DX3631 single $150 Low  ED3645 single $150 Low  II1217 single $150 Low  JB9081 single $150 Low  RP7455 single $150 Low  YC6034 single $150 Low  ZG4257 single $150 Low  DE1048 single $150 Low  ED3645 single $150 Low  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  QJ4248 triple $340 Low  QZ0211 triple $340 Low  RN0807 triple $340 Low  SA4741 triple $340 Low  ST0707 triple $340 Low  XO7284 triple $340 Low  YI4936 triple $340 Low  DE1048 triple $340 Low  DX3631 triple $340 Low  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  EI1464 triple $340 Low  EQ5799 triple $340 Low  ER4237 triple $340 Low  KW1189 triple $340 Low  ML4496 triple $340 Low  PB3773 triple $340 Low  PN6518 triple $340 Low  RK0542 triple $340 Low  RO5385 triple $340 Low  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  RP7455 triple $340 Low  SV8768 triple $340 Low  XP6803 triple $340 Low  BW1653 quad $450 Medium  DE1048 quad $450 Medium  DX7910 quad $450 Medium  FA3216 quad $450 Medium  GT8864 quad $450 Medium  II1217 quad $450 Medium  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  LD3924 quad $450 Medium  MA8181 quad $450 Medium  ND6602 quad $450 Medium  OE0070 quad $450 Medium  PN6518 quad $450 Medium  QJ4248 quad $450 Medium  RS4358 quad $450 Medium  WQ9489 quad $450 Medium  YN0014 quad $450 Medium  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  BU6588 quad $450 Medium  DE1048 quad $450 Medium  DX3631 quad $450 Medium  GT8864 quad $450 Medium  PN6518 quad $450 Medium  PQ7318 quad $450 Medium  RN0807 quad $450 Medium  RT2938 quad $450 Medium  YN0014 quad $450 Medium  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  PB3773 quad $450 Medium  BF5792 quad $450 Medium  BP6369 quad $450 Medium  DE1048 quad $450 Medium  EI1464 quad $450 Medium  LR4975 quad $450 Medium  OE0070 quad $450 Medium  YD7962 quad $450 Medium  YN0014 quad $450 Medium    The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  RP7455 queen $600 Medium  VC1438 queen $600 Medium  VH4324 queen $600 Medium  DX7910 queen $600 Medium  II1217 queen $600 Medium  IQ2935 queen $600 Medium  LR4975 queen $600 Medium  PF9600 queen $600 Medium  QQ9992 queen $600 Medium  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  RI6085 queen $600 Medium  UK7117 queen $600 Medium  VH4324 queen $600 Medium  VJ9868 queen $600 Medium  BU6588 queen $600 Medium  DX7910 queen $600 Medium  EL6562 queen $600 Medium  EX0916 queen $600 Medium  IM4245 queen $600 Medium  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  IQ2935 queen $600 Medium  NE9884 queen $600 Medium  PB3773 queen $600 Medium  XD4686 queen $600 Medium  ZJ4688 queen $600 Medium  BF6818 king $800 High  BH5341 king $800 High  DD4468 king $800 High  EQ5799 king $800 High  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  EX0916 king $800 High  GW6365 king $800 High  HR4409 king $800 High  IM4245 king $800 High  NE9884 king $800 High  PB3773 king $800 High  RP7455 king $800 High  XD4686 king $800 High  XI1297 king $800 High  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  FA3216 king $800 High  FS0685 king $800 High  HL8317 king $800 High  IM4245 king $800 High  VL3025 king $800 High  BH5341 king $800 High  DD4468 king $800 High  FS0685 king $800 High  GW6365 king $800 High  The price of Reserved ROOM  Reservation ID ROOM\_TYPE ROOMP PRICE  -------------------- --------- ----- ------  MX9588 king $800 High  QZ0211 king $800 High  YC6034 king $800 High |

7.3.1

Purpose : return payment id and name only to reduce interface clutter for experienced employees who understand and recognize each payment method by their name so no need read description.

SQL

|  |
| --- |
| CREATE or replace VIEW PaymentName as  select distinct P.payment\_id, payment\_name  from Payment\_method P, Reservation R, Dining D, Meal M  where M.meal\_id=&ENUMBER and P.Payment\_id = R.Payment\_id and R.Reservation\_id = D.Reservation\_id and D.meal\_id = M.meal\_id  order by p.payment\_id； |

Output

|  |
| --- |
| Payment ID Payment Name  ---------- -------------------------  1 TNG eWallet  2 Master Card  3 Visa  4 PayPal  5 FPX  6 Alipay  7 Cash  8 American Express  9 telegraphic transfer  10 letter of credit |

7.3.2

Purpose: return reservation id and guest check out date only so that staff can quickly identify when guest checkout as screen output contains only relevant details by calling one line of sql code so can reduce time taken.

SQL

|  |
| --- |
| ACCEPT ENUMBER PROMPT 'Guest ID. :'  CREATE or replace VIEW GuestCheckoutDate as  SELECT R.guest\_id, R.reservation\_id, CheckOutDate  FROM CheckOut C, Guest G, reservation R  WHERE G.guest\_id=R.guest\_id and R.CheckOutID = C.CheckOutID and R.guest\_id=&ENUMBER  order by checkOutDate desc; |

Output

|  |
| --- |
| Guest ID Reservation ID Check Out Da  -------- ------------------------------ ------------  1001 YI4936 11-JUL-22  1001 EI1464 11-MAR-22  1001 BP6369 07-SEP-16  1001 ER4237 05-OCT-15  1001 GT8864 06-AUG-15 |

7.4.1 View for service table with total price

SQL

|  |
| --- |
| COLUMN service\_id heading 'ID' FORMAT A5  COLUMN service\_name heading 'Name' FORMAT A35  COLUMN service\_descr heading 'Description' FORMAT A35  COLUMN service\_price heading 'Price(RM)' FORMAT 999.99  COLUMN COUNT(R.service\_id) heading 'Number of Services' FORMAT 99  COLUMN SUM(service\_price) heading 'Total Service Price(RM)' FORMAT 999,999.99  TTITLE LEFT 'SERVICE TABLE ' -  RIGHT 'Page: ' FORMAT 999 sql.pno skip 2  SET PAGESIZE 200  SET LINESIZE 200  CREATE or REPLACE view service\_table as  SELECT S.service\_id, service\_name, service\_descr, service\_price, COUNT(R.service\_id) AS "Number of Services", SUM(service\_price) AS "Total Service Price(RM)"  FROM service S, service\_record R  WHERE S.service\_id = R.service\_id  GROUP BY S.service\_id, service\_name, service\_descr, service\_price  ORDER BY service\_price; |

Output

|  |
| --- |
| SERVICE TABLE  Page: 1  ID Name Description Price(RM) Number of Services Total Service Price(RM)  ----- ----------------------------------- ----------------------------------- --------- ------------------ -----------------------  L1214 Room service order food and beverages 10.00 22 220  F6126 In-room amenities 23.99 25 599.75  D6253 Laundry and dry-cleaning services 25.99 36 935.64  G5494 Business Centre office equipment for guests to use 34.59 34 1176.06  V8040 Housekeeping 49.99 33 1649.67  E6997 Front desk and concierge services check-in and check-out 50.00 37 1850  B6201 Shuttle services 88.99 25 2224.75  W1729 Fitness centre 359.98 36 12959.28  V1322 Spa 429.99 32 13759.68  G1852 Meeting and event spaces 899.88 20 17997.6 |

7.4.2 View for service record group by year with total price

SQL

|  |
| --- |
| COLUMN service\_id heading 'ID' FORMAT A5  COLUMN service\_name heading 'Name' FORMAT A35  COLUMN service\_date heading 'Date' FORMAT A11  COLUMN SUM(service\_price) heading 'Total Service Price(RM)' FORMAT 999,999.99  TTITLE LEFT 'SERVICE RECORD TABLE YEAR ' -  RIGHT 'Page: ' FORMAT 999 sql.pno skip 2  SET PAGESIZE 300  SET LINESIZE 64  CREATE or REPLACE view service\_yearly\_totals as  SELECT S.service\_id, service\_name, EXTRACT(YEAR FROM service\_date) AS YEAR, SUM(service\_price) AS "Total Price"  FROM service S, service\_record R  WHERE S.service\_id = R.service\_id  GROUP BY S.service\_id, service\_name, EXTRACT(YEAR FROM service\_date)  ORDER BY EXTRACT(YEAR FROM service\_date); |

Output

|  |
| --- |
| SERVICE RECORD TABLE YEAR Page: 1  ID Name YEAR Total Price  ----- ----------------------------------- ---------- -----------  G5494 Business Centre 2015 172.95  W1729 Fitness centre 2015 2519.86  E6997 Front desk and concierge services 2015 150  V8040 Housekeeping 2015 49.99  F6126 In-room amenities 2015 47.98  D6253 Laundry and dry-cleaning services 2015 103.96  L1214 Room service 2015 10  B6201 Shuttle services 2015 177.98  V1322 Spa 2015 1289.97  G5494 Business Centre 2016 69.18  W1729 Fitness centre 2016 2159.88  E6997 Front desk and concierge services 2016 50  V8040 Housekeeping 2016 49.99  F6126 In-room amenities 2016 23.99  D6253 Laundry and dry-cleaning services 2016 77.97  G1852 Meeting and event spaces 2016 1799.76  L1214 Room service 2016 20  B6201 Shuttle services 2016 444.95  V1322 Spa 2016 859.98  G5494 Business Centre 2017 138.36  W1729 Fitness centre 2017 359.98  E6997 Front desk and concierge services 2017 400  V8040 Housekeeping 2017 199.96  F6126 In-room amenities 2017 143.94  D6253 Laundry and dry-cleaning services 2017 103.96  G1852 Meeting and event spaces 2017 5399.28  L1214 Room service 2017 30  B6201 Shuttle services 2017 266.97  V1322 Spa 2017 1719.96  G5494 Business Centre 2018 172.95  W1729 Fitness centre 2018 1799.9  E6997 Front desk and concierge services 2018 300  V8040 Housekeeping 2018 449.91  F6126 In-room amenities 2018 47.98  D6253 Laundry and dry-cleaning services 2018 181.93  G1852 Meeting and event spaces 2018 1799.76  L1214 Room service 2018 20  B6201 Shuttle services 2018 533.94  V1322 Spa 2018 2579.94  G5494 Business Centre 2019 207.54  W1729 Fitness centre 2019 1439.92  E6997 Front desk and concierge services 2019 200  V8040 Housekeeping 2019 199.96  F6126 In-room amenities 2019 47.98  D6253 Laundry and dry-cleaning services 2019 25.99  G1852 Meeting and event spaces 2019 2699.64  L1214 Room service 2019 30  B6201 Shuttle services 2019 177.98  V1322 Spa 2019 3009.93  G5494 Business Centre 2020 242.13  W1729 Fitness centre 2020 1439.92  E6997 Front desk and concierge services 2020 350  V8040 Housekeeping 2020 99.98  F6126 In-room amenities 2020 95.96  D6253 Laundry and dry-cleaning services 2020 77.97  G1852 Meeting and event spaces 2020 3599.52  L1214 Room service 2020 30  B6201 Shuttle services 2020 177.98  V1322 Spa 2020 859.98  G5494 Business Centre 2021 103.77  W1729 Fitness centre 2021 1439.92  E6997 Front desk and concierge services 2021 300  V8040 Housekeeping 2021 549.89  F6126 In-room amenities 2021 71.97  D6253 Laundry and dry-cleaning services 2021 207.92  G1852 Meeting and event spaces 2021 1799.76  L1214 Room service 2021 50  B6201 Shuttle services 2021 88.99  V1322 Spa 2021 2579.94  G5494 Business Centre 2022 34.59  W1729 Fitness centre 2022 1439.92  E6997 Front desk and concierge services 2022 50  V8040 Housekeeping 2022 49.99  F6126 In-room amenities 2022 71.97  D6253 Laundry and dry-cleaning services 2022 77.97  G1852 Meeting and event spaces 2022 899.88  L1214 Room service 2022 20  B6201 Shuttle services 2022 266.97  V1322 Spa 2022 859.98  G5494 Business Centre 2023 34.59  W1729 Fitness centre 2023 359.98  E6997 Front desk and concierge services 2023 50  F6126 In-room amenities 2023 47.98  D6253 Laundry and dry-cleaning services 2023 77.97  L1214 Room service 2023 10  B6201 Shuttle services 2023 88.99 |

7.4.1 View the total income of meal

SQL

|  |
| --- |
| column meal\_id heading 'Meal ID' format 99  column meal\_name heading 'Meal Name' format a30  column total\_income heading 'Total Income' format 999,999.99  TTITLE Center 'Total Income of Meal In The Year'  create or replace view income\_of\_meal as  SELECT d.meal\_id, meal\_name, sum(meal\_price\*meal qunatity) as total\_income  FROM meal m, dining d  WHERE m.meal\_id = d.meal\_id  GROUP by d.meal\_id, meal\_name  order by d.meal\_id; |

Output

|  |
| --- |
| Total Income of Meal  Meal ID Meal Name Total Income  ------- ------------------------------ ------------  1 Garlic Bread 2,258.42  2 Noodle 197,508.20  3 Egg Fried Rice 14,348.91  4 Chicken Chop 11,965.88  5 Chicken Burger 10,768.65  6 Chicken Sandwitch 79,433.40  7 Salad 65,543.39  8 Muffin 9,448.07  9 Banana Ice cream 100,590.42  10 Hazelnut cake 51,257.15 |

7.4.2 View the Quantity of Meal Ordered

SQL

|  |
| --- |
| column reservation\_id heading 'Reservation ID' format A15  column meal\_id heading 'Meal ID' format 99  column meal\_name heading 'Meal Name' format a30  column meal\_quantity heading 'Meal Ordered' format 999  TTITLE Center 'Quantity of Meal Ordered' -  RIGHT 'Page:' FORMAT 999 SQL.PNO SKIP 2  create or replace view meal\_popularity as  SELECT d.reservation\_id, d.meal\_id, m.meal\_name, d.meal\_quantity,  CASE  WHEN meal\_quantity > 8 THEN 'Popular'  WHEN meal\_quantity > 5 THEN 'Decent'  ELSE 'Not Popular'  END AS Popularity  FROM dining d, meal m, reservation R  WHERE m.meal\_id = d.meal\_id and R.reservation\_id = d.reservation\_id; |

Output

|  |
| --- |
| Quantity of Meal Ordered in this year Page: 1  Reservation ID Meal ID Meal Name Meal Ordered POPULARITY  --------------- ------- ------------------------------ ------------ -----------  AZ1911 1 Garlic Bread 8 Decent  DE1048 1 Garlic Bread 1 Not Popular  DF4057 1 Garlic Bread 7 Decent  ED3645 1 Garlic Bread 10 Popular  ER4237 1 Garlic Bread 1 Not Popular  ER4237 1 Garlic Bread 10 Popular  GW6365 1 Garlic Bread 9 Popular  HL8317 1 Garlic Bread 9 Popular  LD3924 1 Garlic Bread 8 Decent  LV2501 1 Garlic Bread 5 Not Popular |