Технически университет – Варна

Курсов проект по дисциплина:

"**Програмиране за бази от данни**"

Изготвил: Тодор Иванов Йорданов Ф.№ 20621600, СИТ, III курс, 4^a група

Съдържание

1.Задание

Задание ПБД   **8**

Да се проектира и реализира база от данни за **ХОТЕЛ**, която да съхранява следната информация:

- *Клиент* - име, адрес, телефон;
- *Служител* - име, позиция, телефон;
- *Стая* – номер, вид, цена
- *Резервация* - клиент, служител, стая, дата, период;

Правила:

- Всеки клиент може да има повече от една резервация, в различно време,
- Всяка стая може да има само един клиент по всяко време,

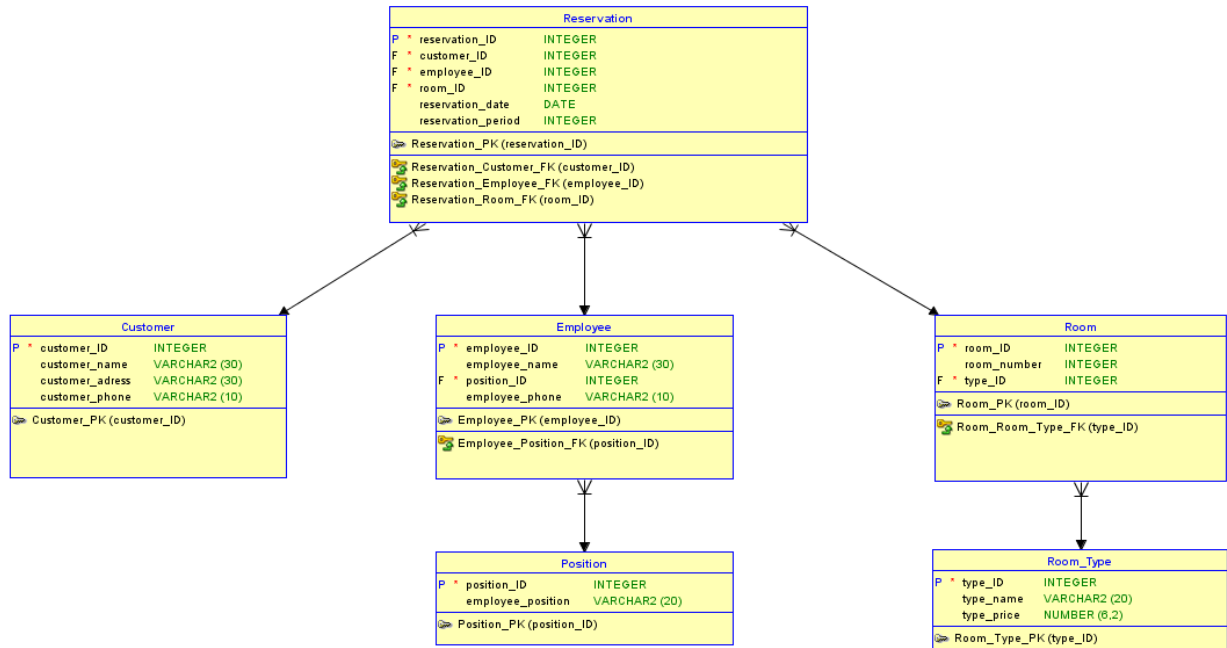Базата от данни трябва да е **НОРМАЛИЗИРАНА** и да позволява:
1. **Въвеждане и корекции** на данни.
2. **Търсене/резервиране** на стая по: тип, дата, период.
3. **Справки** за:
- обща сметка на клиент;
- свободни стаи за тип/дата/период;
- резервации за служител, подредени по дата.

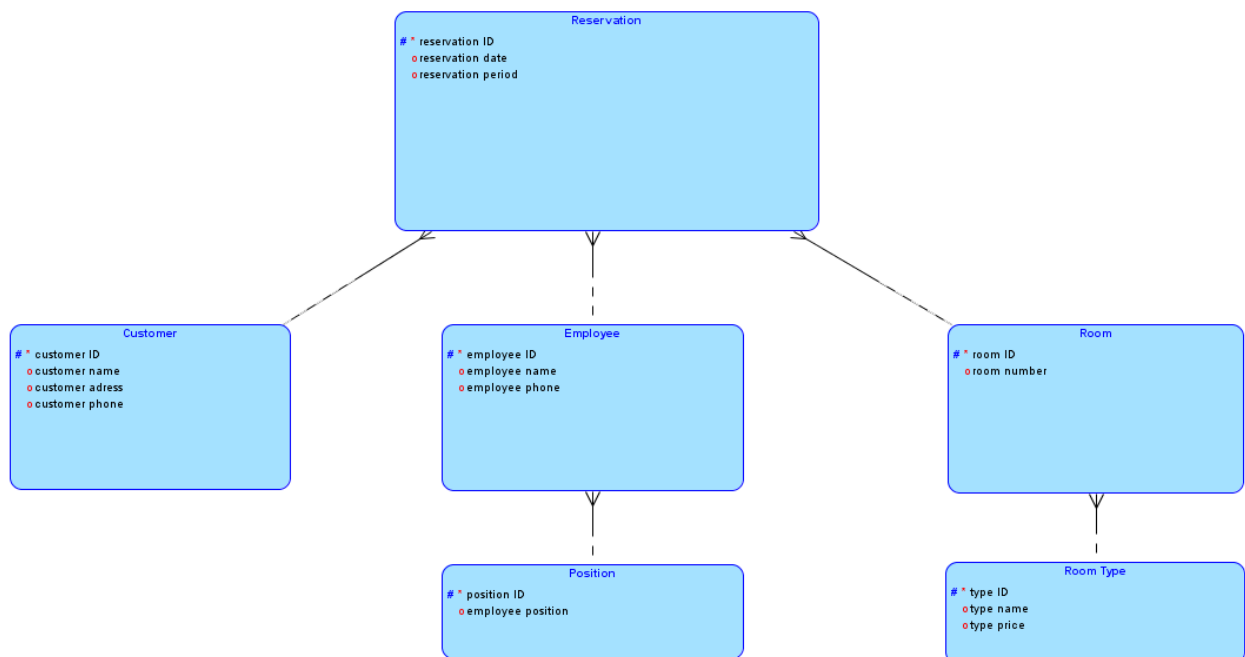Документацията към реализирания проект трябва да съдържа:
- Задание
- Модели (Oracle Data Modeler)   4 седм./10 т.
- SQL команди – DDL, DML   7 седм./5 т.
- PL/SQL процедури/тригери/курсори   13 седм./25 т.
                                        докум./10 т.

2.Модели

Релационен модел

Reservation

P	*	reservation_ID	INTEGER
F	*	customer_ID	INTEGER
F	*	employee_ID	INTEGER
F	*	room_ID	INTEGER
		reservation_date	DATE
		reservation_period	INTEGER

Reservation_PK (reservation_ID)

Reservation_Customer_FK (customer_ID)
Reservation_Employee_FK (employee_ID)
Reservation_Room_FK (room_ID)

Customer

P	*	customer_ID	INTEGER
		customer_name	VARCHAR2 (30)
		customer_adress	VARCHAR2 (30)
		customer_phone	VARCHAR2 (10)

Customer_PK (customer_ID)

Employee

P	*	employee_ID	INTEGER
		employee_name	VARCHAR2 (30)
F	*	position_ID	INTEGER
		employee_phone	VARCHAR2 (10)

Employee_PK (employee_ID)

Employee_Position_FK (position_ID)

Room

P	*	room_ID	INTEGER
		room_number	INTEGER
F	*	type_ID	INTEGER

Room_PK (room_ID)

Room_Room_Type_FK (type_ID)

Position

P	*	position_ID	INTEGER
		employee_position	VARCHAR2 (20)

Position_PK (position_ID)

Room_Type

P	*	type_ID	INTEGER
		type_name	VARCHAR2 (20)
		type_price	NUMBER (6,2)

Room_Type_PK (type_ID)

Логически модел

Reservation

* reservation ID
o reservation date
o reservation period

Customer

* customer ID
o customer name
o customer adress
o customer phone

Employee

* employee ID
o employee name
o employee phone

Room

* room ID
o room number

Position

* position ID
o employee position

Room Type

* type ID
o type name
o type price

3.SQL команди(DDL,DML)

DDL

--Създаавам таблица за клиенти

CREATE TABLE customer (

   customer_id   INTEGER NOT NULL ,

   customer_name   VARCHAR2(30),

   customer_adress VARCHAR2(30),

   customer_phone   VARCHAR2(10)

   CONSTRAINT customer_phone_length CHECK (LENGTHB(customer_phone) = 10),

   CONSTRAINT customer_id_rule CHECK (customer_id > 0)

);

ALTER TABLE customer ADD CONSTRAINT customer_pk PRIMARY KEY (customer_id);

--Създавам таблица за служители

CREATE TABLE employee (

   employee_id   INTEGER NOT NULL,

   employee_name   VARCHAR2(30),

   position_id   INTEGER NOT NULL,

   employee_phone VARCHAR2(10)

   CONSTRAINT employee_phone_length CHECK (LENGTHB(employee_phone) = 10),

   CONSTRAINT employee_id_rule CHECK (employee_id > 0)

);

ALTER TABLE employee ADD CONSTRAINT employee_pk PRIMARY KEY (employee_id);

--Създавам таблица за позиции на служители

CREATE TABLE positions (

   position_id   INTEGER NOT NULL,

   employee_position VARCHAR2(20),

```sql
    CONSTRAINT position_id_rule CHECK (position_id > 0)
);


ALTER TABLE positions ADD CONSTRAINT position_pk PRIMARY KEY ( position_id );


--Създавам таблица за резервации
CREATE TABLE reservation (
    reservation_id     INTEGER NOT NULL,
    customer_id        INTEGER NOT NULL,
    employee_id        INTEGER NOT NULL,
    room_id            INTEGER NOT NULL,
    reservation_date    DATE,
    reservation_period   INTEGER
    CONSTRAINT res_period_rule CHECK(reservation_period > 0),
    CONSTRAINT res_id_rule CHECK (reservation_id > 0)
);


ALTER TABLE reservation ADD CONSTRAINT reservation_pk PRIMARY KEY ( reservation_id );


--Създавам таблиа за стаи
CREATE TABLE room (
    room_id      INTEGER NOT NULL,
    room_number   INTEGER,
    type_id      INTEGER NOT NULL,
    CONSTRAINT room_number_rule CHECK(room_number > 0),
    CONSTRAINT room_id_rule CHECK (room_id > 0)
);


ALTER TABLE room ADD CONSTRAINT room_pk PRIMARY KEY ( room_id );


--Създавам таблици за тип стаи
CREATE TABLE room_type (
    type_id     INTEGER NOT NULL,
```

```
    type_name   VARCHAR2(20),

    type_price   NUMBER(6,2),

    CONSTRAINT type_price_rule CHECK(type_price > 0),

    CONSTRAINT type_id_rule CHECK (type_id > 0)
);


ALTER TABLE room_type ADD CONSTRAINT room_type_pk PRIMARY KEY ( type_id );



--Задавам foreign keys пораждащи връзките между отделните таблици
ALTER TABLE employee
    ADD CONSTRAINT employee_position_fk FOREIGN KEY ( position_id )
        REFERENCES positions ( position_id ) ON DELETE CASCADE;


ALTER TABLE reservation
    ADD CONSTRAINT reservation_customer_fk FOREIGN KEY ( customer_id )
        REFERENCES customer ( customer_id ) ON DELETE CASCADE;


ALTER TABLE reservation
    ADD CONSTRAINT reservation_employee_fk FOREIGN KEY ( employee_id )
        REFERENCES employee ( employee_id ) ON DELETE CASCADE;


ALTER TABLE reservation
    ADD CONSTRAINT reservation_room_fk FOREIGN KEY ( room_id )
        REFERENCES room ( room_id ) ON DELETE CASCADE;


ALTER TABLE room
    ADD CONSTRAINT room_room_type_fk FOREIGN KEY ( type_id )
        REFERENCES room_type ( type_id ) ON DELETE CASCADE;
```

DML

--добавяне на информация в таблица Customer

insert into Customer values(1,'Todor Todorov','BG,Varna,st.№4,ap.№16','0878122134');

insert into Customer values(2,'Ivan Ivanov','BG,Vratsa,st.№22,ap.№1','0883548194');

insert into Customer values(3,'Petq Petrova','BG,Sofia,st.№348,ap.№30','0898781234');

insert into Customer values(4,'Katq Jeleva','BG,Plovdiv,st.№1,ap.№1','0882998126');

insert into Customer values(5,'Peter Georgiev','BG,Burgas,st.№8,ap.№7','0893748555');

insert into Customer values(6,'Kristiqn Tomov','BG,Smolqn,st.№11,ap.№20','0882778554');

insert into Customer values(7,'Vasil Minov','BG,Pernik,st.№3,ap.№5','0881548111');

insert into Customer values(8,'Elitsa Krusteva','BG,Dobrich,st.№1,ap.№7','0873739197');

insert into Customer values(9,'Kalina Kirilova','BG,Vidin,st.№9,ap.№9','0893341144');

insert into Customer values(10,'Simona Pencheva','BG,Petrich,st.№12,ap.№8','0882522117');

insert into Customer values(11,'Simeon Tanev','BG,Varna,st.№9,ap.№13','0878548878');

insert into Customer values(12,'Maria Toteva','BG,Plovdiv,st.№2,ap.№2','0888848194');

insert into Customer values(13,'Andreq Maneva','BG,Sofia,st.№44,ap.№14','0883535355');

insert into Customer values(14,'Silviq Vasileva','BG,Karnobat,st.№4,ap.№8','0882901345');

insert into Customer values(15,'Galin Plamenov','BG,Burgas,st.№55,ap.№25','0883131366');

insert into Customer values(16,'Viktoriq Ivanova','BG,Sofia,st.№15,ap.№15','0881515193');

insert into Customer values(17,'Milen Todorov','BG,Pernik,st.№14,ap.№28','0891248112');

insert into Customer values(18,'Dobri Dobrev','BG,Varna,st.№51,ap.№17','0873578118');

insert into Customer values(19,'Malina Todorova','BG,Kalofer,st.№6,ap.№7','0877541452');

insert into Customer values(20,'Zahari Dobrev','BG,Haskovo,st.№5,ap.№14','0886566191');

insert into Customer values(21,'Elon Musk','USA,Texas,st.№75,h.№443','0884448191');

update customer

set customer_name = 'Elitsa Encheva'

WHERE customer_name = 'Elitsa Krusteva';

delete from Customer

where customer_name ='Elon Musk';

insert into Customer values(21,'Elon Musk','USA,Texas,st.№75,h.№443','0884448191');

```
--Добавяне на информация в таблица room_type
insert into room_type values(1,'single',100.00);
insert into room_type values(2,'double',200.00);
insert into room_type values(3,'triple',300.00);
insert into room_type values(4,'quad',400.00);
insert into room_type values(5,'apartment',500.00);
insert into room_type values(6,'president suite',1000.00);


update room_type
set type_price = 2000
where type_name = 'president suite';




--Добавяне на информация в таблица room
insert into room values(1,101,1);
insert into room values(2,102,2);
insert into room values(3,103,3);
insert into room values(4,104,4);
insert into room values(5,105,5);
insert into room values(6,201,1);
insert into room values(7,202,2);
insert into room values(8,203,3);
insert into room values(9,204,4);
insert into room values(10,205,5);
insert into room values(11,301,1);
insert into room values(12,302,2);
insert into room values(13,303,3);
insert into room values(14,304,4);
insert into room values(15,305,5);
```

```
insert into room values(16,401,1);

insert into room values(17,402,2);

insert into room values(18,403,3);

insert into room values(19,404,4);

insert into room values(20,405,5);

insert into room values(21,501,1);

insert into room values(22,502,2);

insert into room values(23,503,3);

insert into room values(24,504,4);

insert into room values(25,505,5);

insert into room values(26,601,6);


delete from room

WHERE room_id =26;


insert into room values(26,601,1);

insert into room values(27,602,2);

insert into room values(28,603,3);

insert into room values(29,604,4);

insert into room values(30,605,5);

insert into room values(31,701,1);

insert into room values(32,702,2);

insert into room values(33,703,3);

insert into room values(34,704,4);

insert into room values(35,705,5);

insert into room values(36,801,1);

insert into room values(37,802,2);

insert into room values(38,803,3);

insert into room values(39,804,4);

insert into room values(40,805,5);

insert into room values(41,901,6);
```

--Добавяне на информация в таблица positions

insert into positions values(1,'manager');

insert into positions values(2,'receptionist');

insert into positions values(3,'porter');

insert into positions values(4,'host');

insert into positions values(5,'hostess');

--Добавяне на информация в таблица employee

insert into employee values(1,'Toma Tomov',1,'0882960223');

insert into employee values(2,'Roza Ivanova',5,'0878366213');

insert into employee values(3,'Radoslav Nikolov',4,'0894735212');

insert into employee values(4,'Teodora Trifonova',2,'0896672873');

insert into employee values(5,'Marin Marinov',3,'0885167654');

insert into employee values(6,'Petq Nikolova',2,'0878211318');

--Добавяне на информация в таблица reservation

insert into Reservation values(1,1,6,3,'2021-01-04',7);--3

insert into Reservation values(2,2,4,12,'2021-01-18',7);--12

insert into Reservation values(3,3,5,22,'2021-01-25',7);--22

insert into Reservation values(4,4,2,7,'2021-02-01',7);--7

insert into Reservation values(5,5,3,15,'2021-02-15',7);--15

insert into Reservation values(6,6,2,18,'2021-03-03',5);--18

insert into Reservation values(7,7,3,24,'2021-03-19',3);--24

insert into Reservation values(8,8,6,37,'2021-04-01',4);--37

insert into Reservation values(9,9,5,29,'2021-04-16',3);--29

insert into Reservation values(10,10,4,33,'2021-05-10',14);--33

insert into Reservation values(11,11,6,9,'2021-06-07',7);--9

insert into Reservation values(12,12,5,40,'2021-06-19',2);--40

insert into Reservation values(13,13,4,26,'2021-07-05',7);--26

insert into Reservation values(14,14,3,20,'2021-07-24',2);--20

insert into Reservation values(15,15,2,16,'2021-08-02',14);--16

insert into Reservation values(16,16,3,31,'2021-09-10',3);--31

insert into Reservation values(17,17,5,39,'2021-09-20',7);--39

insert into Reservation values(18,18,6,5,'2021-10-09',2);--5

insert into Reservation values(19,19,2,35,'2021-10-25',7);--35

insert into Reservation values(20,20,4,28,'2021-11-06',2);--28

insert into Reservation values(21,21,1,41,'2021-07-01',30);--41

--2. Търсене на стая по тип,дата,период (свободни стаи)

SELECT r.room_number,rt.type_name,rr.reservation_date FROM room r

JOIN room_type rt ON r.type_id = rt.type_id

LEFT JOIN reservation rr ON rr.room_id = r.room_id

WHERE rt.type_name LIKE '&Type_name' AND (rr.room_id IS NULL OR ((rr.reservation_date +
rr.reservation_period) < to_date('&Reservation_datee','yyyy/mm/dd') OR rr.reservation_date >
(to_date('&Reservation_datee','yyyy/mm/dd') + '&Reservation_periodd')));

--3. Справки за

-- Обща сметка на клиент

SELECT c.customer_name, r.reservation_id, r.reservation_date, SUM(r.reservation_period*rt.type_price)
"TOTAL BILL" FROM reservation r

JOIN customer c ON r.customer_id = c.customer_id

JOIN room rm ON r.room_id = rm.room_id

JOIN room_type rt ON rm.type_id = rt.type_id

WHERE c.customer_name = '&cus_name'

Group BY c.customer_name, r.reservation_id, r.reservation_date;

-- Свободни стаи за тип/дата/период

SELECT r.room_number,rt.type_name,rr.reservation_date FROM room r

JOIN room_type rt ON r.type_id = rt.type_id

LEFT JOIN reservation rr ON rr.room_id = r.room_id

WHERE rt.type_name LIKE '&Type_name' AND (rr.room_id IS NULL OR ((rr.reservation_date + rr.reservation_period) < to_date('&Reservation_datee','yyyy/mm/dd') OR rr.reservation_date > (to_date('&Reservation_datee','yyyy/mm/dd') + '&Reservation_periodd')));

-- Резервации за служител подредени по дата

SELECT e.employee_name, r.reservation_id, c.customer_name, r.reservation_date,rm.room_number, r.reservation_period FROM reservation r

JOIN employee e ON r.employee_id = e.employee_id

JOIN customer c ON r.customer_id = c.customer_id

JOIN room rm ON r.room_id = rm.room_id

WHERE e.employee_name = '&empl_name'

ORDER BY r.reservation_date;

4.PL/SQL (процедури/тригери/курсори)

---------Sequence и Triggers за auto increment на ID-та при попълване на данни в таблиците---------

CREATE SEQUENCE positions_seq START WITH 5;

CREATE OR REPLACE TRIGGER positions_id_auto_trigger

BEFORE INSERT ON positions FOR EACH ROW WHEN (NEW.position_id IS NULL)

BEGIN

  :NEW.position_id := positions_seq.NEXTVAL;

END;

```
CREATE SEQUENCE room_type_seq START WITH 6;

CREATE OR REPLACE TRIGGER room_type_id_auto_trigger

BEFORE INSERT ON room_type FOR EACH ROW WHEN (NEW.type_id IS NULL)

BEGIN

   :NEW.type_id := room_type_seq.NEXTVAL;

END;



CREATE SEQUENCE customer_seq START WITH 21;

CREATE OR REPLACE TRIGGER customer_id_auto_trigger

BEFORE INSERT ON customer FOR EACH ROW WHEN (NEW.customer_id IS NULL)

BEGIN

   :NEW.customer_id := customer_seq.NEXTVAL;

END;



CREATE SEQUENCE employee_seq START WITH 6;

CREATE OR REPLACE TRIGGER employee_id_auto_trigger

BEFORE INSERT ON employee FOR EACH ROW WHEN (NEW.employee_id IS NULL)

BEGIN

   :NEW.employee_id := employee_seq.NEXTVAL;

END;



CREATE SEQUENCE room_seq START WITH 41;

CREATE OR REPLACE TRIGGER room_id_auto_trigger

BEFORE INSERT ON room FOR EACH ROW WHEN (NEW.room_id IS NULL)

BEGIN

   :NEW.room_id := room_seq.NEXTVAL;

END;
```

```sql
CREATE SEQUENCE reservation_seq START WITH 21;

CREATE OR REPLACE TRIGGER reservation_id_auto_trigger

BEFORE INSERT ON reservation FOR EACH ROW WHEN (NEW.reservation_id IS NULL)

BEGIN

   :NEW.reservation_id := reservation_seq.NEXTVAL;

END;
```

```sql
--Constraint за уникална позиция
ALTER TABLE positions
ADD CONSTRAINT ep_unique UNIQUE (employee_position);


--Constraint за уникален телефонен номер на служител
ALTER TABLE employee
ADD CONSTRAINT eph_unique UNIQUE (employee_phone);


--Constraint за уникален телефонен номер на клиент
ALTER TABLE customer
ADD CONSTRAINT cph_unique UNIQUE (customer_phone);


ALTER TABLE customer
MODIFY customer_adress VARCHAR2( 255 ) NOT NULL;


ALTER TABLE room_type
ADD CONSTRAINT type_unique UNIQUE (type_name);


ALTER TABLE room
ADD CONSTRAINT room_unique UNIQUE (room_number);
```

---------------Процедури за изпълнение при въвеждане на данни в таблиците------------

```
CREATE OR REPLACE PROCEDURE InsertPosition(

in_position_name IN positions.employee_position%TYPE

) IS

BEGIN

    INSERT INTO positions(employee_position) values(in_position_name);

END;
```

```
CREATE OR REPLACE PROCEDURE InsertEmployee(

in_employee_name IN employee.employee_name%TYPE,

in_position_id IN employee.position_id%TYPE,

in_employee_phone IN employee.employee_phone%TYPE

) IS

BEGIN

    INSERT INTO employee(employee_name, position_id, employee_phone) values(in_employee_name,
in_position_id, in_employee_phone);

END;
```

```
CREATE OR REPLACE PROCEDURE InsertCustomer(

in_customer_name IN customer.customer_name%TYPE,

in_customer_adress IN customer.customer_adress%TYPE,

in_customer_phone IN customer.customer_phone%TYPE

) IS

BEGIN

    INSERT INTO customer(customer_name, customer_adress, customer_phone) values(in_customer_name,
in_customer_adress, in_customer_phone);

END;
```

```
CREATE OR REPLACE PROCEDURE InsertRoomType(
```

```
in_type_name IN room_type.type_name%TYPE,

in_type_price IN room_type.type_price%TYPE

) IS

BEGIN

    INSERT INTO room_type(type_name, type_price) values(in_type_name, in_type_price);

END;




CREATE OR REPLACE PROCEDURE InsertRoom(

in_room_number IN room.room_number%TYPE,

in_type_id IN room.type_id%TYPE

) IS

BEGIN

    INSERT INTO room(room_number, type_id) values(in_room_number, in_type_id);

END;




CREATE OR REPLACE PROCEDURE InsertReservation(

in_customer_id IN reservation.customer_id%TYPE,

in_employee_id IN reservation.employee_id%TYPE,

in_room_id IN reservation.room_id%TYPE,

in_reservation_date IN reservation.reservation_date%TYPE,

in_reservation_period IN reservation.reservation_period%TYPE

) IS

BEGIN

    INSERT INTO reservation(customer_id, employee_id, room_id, reservation_date, reservation_period)
values(in_customer_id,in_employee_id,in_room_id,in_reservation_date,in_reservation_period);

END;

-------------------------------------------------------------------------------------


---------------Процедури за изпълнение при ъпдейтване на данни в таблиците------------


CREATE OR REPLACE PROCEDURE UpdatePosition(
```

```sql
in_position_id IN positions.position_id%TYPE,

in_position_name IN positions.employee_position%TYPE

) IS

BEGIN

    UPDATE positions SET employee_position = in_position_name

    WHERE position_id = in_position_id;

END;




CREATE OR REPLACE PROCEDURE UpdateEmployee(

in_employee_id IN employee.employee_id%TYPE,

in_employee_name IN employee.employee_name%TYPE,

in_position_id IN employee.position_id%TYPE,

in_employee_phone IN employee.employee_phone%TYPE

) IS

BEGIN

    UPDATE employee SET employee_name=in_employee_name, position_id=in_position_id,
employee_phone=in_employee_phone

    WHERE employee_id=in_employee_id;

END;




CREATE OR REPLACE PROCEDURE UpdateCustomer(

in_customer_id IN customer.customer_id%TYPE,

in_customer_name IN customer.customer_name%TYPE,

in_customer_adress IN customer.customer_adress%TYPE,

in_customer_phone IN customer.customer_phone%TYPE

) IS

BEGIN

    UPDATE customer SET customer_name=in_customer_name, customer_adress=in_customer_adress,
customer_phone=in_customer_phone

    WHERE customer_id=in_customer_id;

END;
```

```sql
CREATE OR REPLACE PROCEDURE UpdateRoomType(

in_type_id IN room_type.type_id%TYPE,

in_type_name IN room_type.type_name%TYPE,

in_type_price IN room_type.type_price%TYPE

) IS

BEGIN

    UPDATE room_type SET type_name=in_type_name, type_price=in_type_price

    WHERE type_id=in_type_id;

END;



CREATE OR REPLACE PROCEDURE UpdateRoom(

in_room_id IN room.room_id%TYPE,

in_room_number IN room.room_number%TYPE,

in_type_id IN room.type_id%TYPE

) IS

BEGIN

    UPDATE room SET room_number=in_room_number, type_id=in_type_id

    WHERE room_id=in_room_id;

END;



CREATE OR REPLACE PROCEDURE UpdateReservation(

in_reservation_id IN reservation.reservation_id%TYPE,

in_customer_id IN reservation.customer_id%TYPE,

in_employee_id IN reservation.employee_id%TYPE,

in_room_id IN reservation.room_id%TYPE,

in_reservation_date IN reservation.reservation_date%TYPE,

in_reservation_period IN reservation.reservation_period%TYPE

) IS

BEGIN
```

UPDATE reservation SET customer_id=in_customer_id, employee_id=in_employee_id, room_id=in_room_id, reservation_date=in_reservation_date, reservation_period=in_reservation_period

WHERE reservation_id=in_reservation_id;

END;

--------------Процедури за изпълнение при изтриване на данни от таблиците-----------

CREATE OR REPLACE PROCEDURE DeletePosition(

in_position_id IN positions.position_id%TYPE

) IS

BEGIN

DELETE FROM positions WHERE position_id = in_position_id;

END;

CREATE OR REPLACE PROCEDURE DeleteEmployee(

in_employee_id IN employee.employee_id%TYPE

) IS

BEGIN

DELETE FROM employee WHERE employee_id = in_employee_id;

END;

CREATE OR REPLACE PROCEDURE DeleteCustomer(

in_customer_id IN customer.customer_id%TYPE

) IS

BEGIN

DELETE FROM customer WHERE customer_id = in_customer_id;

END;

CREATE OR REPLACE PROCEDURE DeleteRoomType(

```
in_type_id IN room_type.type_id%TYPE

) IS

BEGIN

    DELETE FROM room_type WHERE type_id = in_type_id;

END;




CREATE OR REPLACE PROCEDURE DeleteRoom(

in_room_id IN room.room_id%TYPE

) IS

BEGIN

    DELETE FROM room WHERE room_id = in_room_id;

END;




CREATE OR REPLACE PROCEDURE DeleteReservation(

in_reservation_id IN reservation.reservation_id%TYPE

) IS

BEGIN

    DELETE FROM reservation WHERE reservation_id = in_reservation_id;

END;

------------------------------------------------------------------------------------
```

-----------------Процедура за изпълнение на търсене/резервиране---------------------

--Функция за намиране номер на стая по зададено id(Връща номер на стая)

```
CREATE FUNCTION SearchRoomNumberReservation  (in_room_id IN INTEGER)

  RETURN INTEGER

  IS r_id INTEGER;

    r_number INTEGER;

BEGIN




 SELECT r.room_id,r.room_number INTO r_id,r_number  FROM room r
```

```
        WHERE r.room_id=in_room_id;


        RETURN r_number;
END;


--Функция за намиране на id на свободни стаи(Връща id на първата свободна стая която открие)
CREATE FUNCTION SearchFreeRoom (in_type IN VARCHAR2,in_date IN DATE,in_period IN INTEGER)
  RETURN INTEGER
  IS r_id INTEGER;
    r_number INTEGER;
    r_type VARCHAR2(20);
    r_date DATE;
BEGIN



 SELECT r.room_id,r.room_number,rt.type_name,rr.reservation_date INTO r_id,r_number,r_type,r_date FROM
room r
        JOIN room_type rt ON r.type_id = rt.type_id
        LEFT JOIN reservation rr ON rr.room_id = r.room_id
        WHERE lower(rt.type_name) LIKE lower(in_type) AND (rr.room_id IS NULL OR (  (rr.reservation_date +
rr.reservation_period) < to_date(in_date,'yyyy/mm/dd')  OR  rr.reservation_date >
(to_date(in_date,'yyyy/mm/dd') + in_period )  )) AND ROWNUM <=1;


        RETURN r_id;
END;


--Процедура за резервиране на свободна стая
CREATE OR REPLACE PROCEDURE ReserveRoom(
in_customer_id IN customer.customer_id%TYPE,
in_employee_id IN employee.employee_id%TYPE,
in_room_type IN room_type.type_name%TYPE,
in_date IN reservation.reservation_date%TYPE,
in_period IN reservation.reservation_period%TYPE
) IS
```

```
BEGIN


InsertReservation(in_customer_id,in_employee_id,SearchFreeRoom(in_room_type,in_date,in_period),in_date,
in_period);


    DBMS_OUTPUT.PUT_LINE('Successfully made reservation. {customer_id['||in_customer_id||'],
employee_id['||in_employee_id||'], room type['||in_room_type||'], room
number['||SearchRoomNumberReservation(SearchFreeRoom(in_room_type,in_date,in_period))||'], period
from ['||in_date||'] to ['||(to_date(in_date,'yyyy/mm/dd') + in_period)||'] }');


END;

--EXECUTE ReserveRoom('&customer_id','&employee_id','&room_type','&date','&period');

-----------------------------------------------------------------------------------


-----------------------Процедури за изпълнение на справки---------------------------


CREATE OR REPLACE PROCEDURE QueryCustomerBill(

in_name IN customer.customer_name%TYPE

) IS

BEGIN

  DBMS_OUTPUT.ENABLE;

  FOR i IN (SELECT c.customer_name, r.reservation_id, r.reservation_date,
SUM(r.reservation_period*rt.type_price) "BILL" FROM reservation r

        JOIN customer c ON r.customer_id = c.customer_id

        JOIN room rm ON r.room_id = rm.room_id

        JOIN room_type rt ON rm.type_id = rt.type_id

        WHERE lower(c.customer_name) LIKE lower(in_name)

        Group BY c.customer_name, r.reservation_id, r.reservation_date)

  LOOP

    DBMS_OUTPUT.PUT_LINE('customer:['||i.customer_name||'], reservation_ID:['||i.reservation_id||'],
reservation date:['||i.reservation_date||'], bill:['||i.BILL||'$]');

  END LOOP;

END;

--EXECUTE QueryCustomerBill('&customer_name');
```

```
CREATE OR REPLACE PROCEDURE QueryFreeRooms(

in_type_name IN room_type.type_name%TYPE,

in_date IN reservation.reservation_date%TYPE,

in_period IN reservation.reservation_period%TYPE

) IS

BEGIN

   DBMS_OUTPUT.ENABLE;

   DBMS_OUTPUT.PUT_LINE('----------Free "'||in_type_name||'" rooms from ['||in_date||'] to
['||(to_date(in_date,'yyyy/mm/dd') + in_period)||']----------');

   DBMS_OUTPUT.PUT_LINE('');

   FOR i IN (

       SELECT r.room_number,rt.type_name,rr.reservation_date FROM room r

       JOIN room_type rt ON r.type_id = rt.type_id

       LEFT JOIN reservation rr ON rr.room_id = r.room_id

       WHERE lower(rt.type_name) LIKE lower(in_type_name) AND (rr.room_id IS NULL OR (
(rr.reservation_date + rr.reservation_period) < to_date(in_date,'yyyy/mm/dd')  OR  rr.reservation_date >
(to_date(in_date,'yyyy/mm/dd') + in_period )  ))

       )

   LOOP

     DBMS_OUTPUT.PUT_LINE('                        room number:['||i.room_number||']');

   END LOOP;

     DBMS_OUTPUT.PUT_LINE('');

     DBMS_OUTPUT.PUT_LINE('-----------------------------------------------------------------------');

END;

--EXECUTE QueryFreeRooms('&room_type','&reservation_date','&reservation_period')


CREATE OR REPLACE PROCEDURE QueryEmployeeReservations(

in_name IN employee.employee_name%TYPE

) IS

BEGIN

   DBMS_OUTPUT.ENABLE;

   FOR i IN (

       SELECT e.employee_name, r.reservation_id, c.customer_name, r.reservation_date,rm.room_number,
r.reservation_period FROM reservation r

       JOIN employee e ON r.employee_id = e.employee_id
```

```
            JOIN customer c ON r.customer_id = c.customer_id

            JOIN room rm ON r.room_id = rm.room_id

            WHERE lower(e.employee_name) LIKE lower(in_name)

            ORDER BY r.reservation_date

        )

    LOOP

        DBMS_OUTPUT.PUT_LINE('employee:['||i.employee_name||'], reservation_ID:['||i.reservation_id||'],
customer:['||i.customer_name||'], date:['||i.reservation_date||'], period:['||i.reservation_period||'], room
number:['||i.room_number||']');

    END LOOP;

END;

--EXECUTE QueryEmployeeReservations('&employee_name');

-----------------------------------------------------------------------------------


SET SERVEROUTPUT ON; --Този ред трябва да бъде изпълнен, за да се разреши извеждане
```

5.Примери

Търсене/резервиране на стая по: тип, дата, период.

```
566     --Процедура за резервиране на свободна стая
567  ⊟ CREATE OR REPLACE PROCEDURE ReserveRoom(
568     in_customer_id IN customer.customer_id%TYPE,
569     in_employee_id IN employee.employee_id%TYPE,
570     in_room_type IN room_type.type_name%TYPE,
571     in_date IN reservation.reservation_date%TYPE,
572     in_period IN reservation.reservation_period%TYPE
573     ) IS
574     BEGIN
575
576         InsertReservation(in_customer_id,in_employee_id,SearchFreeRoom(in_room_type,in_date,in_period),in_date,in_period);
577
578         DBMS_OUTPUT.PUT_LINE('Successfully made reservation. {customer_id['||in_customer_id||'], employee_id['||in_employee_id||'], room type['
579
580     END;
581     EXECUTE ReserveRoom('&customer_id','&employee_id','&room_type','&date','&period');
582  ⊟ ----------------------------------------------------------------------------------
583
584     ---------------------Процедури за изпълнение на справки----------------------
```

Script Output X

📌 ✏️ 💾 🖨 📋 | Task completed in 11,439 seconds

```
Successfully made reservation. {customer_id[1], employee_id[1], room type[triple], room number[403], period from [2021-01-01] to [2021-01-11] }


PL/SQL procedure successfully completed.
```

Справка за обща сметка на клиент

```sql
586  CREATE OR REPLACE PROCEDURE QueryCustomerBill(
587  in_name IN customer.customer_name%TYPE
588  ) IS
589  BEGIN
590      DBMS_OUTPUT.ENABLE;
591      FOR i IN (SELECT c.customer_name, r.reservation_id, r.reservation_date, SUM(r.reservation_period*rt.type_price) "BILL" FROM reservation r
592              JOIN customer c ON r.customer_id = c.customer_id
593              JOIN room rm ON r.room_id = rm.room_id
594              JOIN room_type rt ON rm.type_id = rt.type_id
595              WHERE lower(c.customer_name) LIKE lower(in_name)
596              Group BY c.customer_name, r.reservation_id, r.reservation_date)
597      LOOP
598          DBMS_OUTPUT.PUT_LINE('customer:['||i.customer_name||'], reservation_ID:['||i.reservation_id||'], reservation date:['||i.reservation_date||'], bill:['||i.BILL||'$]');
599      END LOOP;
600  END;
601  EXECUTE QueryCustomerBill('&customer_name');
602
```

📌 🖉 💾 🖨 🗔 | Task completed in 7,014 seconds

```
customer:[Elitsa Encheva], reservation_ID:[8], reservation date:[2021-04-01], bill:[800$]


PL/SQL procedure successfully completed.
```

Справка за свободни стаи за тип/дата/период;

```sql
CREATE OR REPLACE PROCEDURE QueryFreeRooms(
in_type_name IN room_type.type_name%TYPE,
in_date IN reservation.reservation_date%TYPE,
in_period IN reservation.reservation_period%TYPE
) IS
BEGIN
    DBMS_OUTPUT.ENABLE;
    DBMS_OUTPUT.PUT_LINE('----------Free "'||in_type_name||'" rooms from ['||in_date||'] to ['||(to_date(in_date,'yyyy/mm/dd') + in_period)||']----------');
    DBMS_OUTPUT.PUT_LINE('');
    FOR i IN (
            SELECT r.room_number,rt.type_name,rr.reservation_date FROM room r
            JOIN room_type rt ON r.type_id = rt.type_id
            LEFT JOIN reservation rr ON rr.room_id = r.room_id
            WHERE lower(rt.type_name) LIKE lower(in_type_name) AND (rr.room_id IS NULL OR (  (rr.reservation_date + rr.reservation_period) < to_date(in_date,'yyyy/mm/dd')
            OR  rr.reservation_date > (to_date(in_date,'yyyy/mm/dd') + in_period )  ))
            )
    LOOP
        DBMS_OUTPUT.PUT_LINE('                          room number:['||i.room_number||']');
    END LOOP;
        DBMS_OUTPUT.PUT_LINE('');
        DBMS_OUTPUT.PUT_LINE('------------------------------------------------------------------');
END;
EXECUTE QueryFreeRooms('&room_type','&reservation_date','&reservation_period')
```

💾 🖨 🗔 | Task completed in 6,011 seconds

```
-----Free "triple" rooms from [2021-01-01] to [2021-01-11]----------

                    room number:[403]
                    room number:[703]
                    room number:[603]
                    room number:[303]
                    room number:[503]
                    room number:[803]
                    room number:[203]


------------------------------------------------------------------
```

Справка за резервации за служител, подредени по дата.

```
625
626 □ CREATE OR REPLACE PROCEDURE QueryEmployeeReservations(
627   in_name IN employee.employee_name%TYPE
628   ) IS
629   BEGIN
630     DBMS_OUTPUT.ENABLE;
631 □   FOR i IN (
632               SELECT e.employee_name, r.reservation_id, c.customer_name, r.reservation_date,rm.room_number, r.reservation_period FROM reservation r
633               JOIN employee e ON r.employee_id = e.employee_id
634               JOIN customer c ON r.customer_id = c.customer_id
635               JOIN room rm ON r.room_id = rm.room_id
636               WHERE lower(e.employee_name) LIKE lower(in_name)
637               ORDER BY r.reservation_date
638             )
639     LOOP
640         DBMS_OUTPUT.PUT_LINE('employee:['||i.employee_name||'], reservation_ID:['||i.reservation_id||'], customer:['||i.customer_name||'], date:['||
641     END LOOP;
642   END;
643   EXECUTE QueryEmployeeReservations('&employee_name');
644   --------------------------------------------------------------------------------------
645
646   SET SERVEROUTPUT ON; --Този ред трябва да бъде изпълнен, за да се разреши извеждане чрез DBMS_OUTPUT
```

Script Output ✕

📌 ✏ 💾 🖨 📋 │ Task completed in 7,001 seconds

```
employee:[Roza Ivanova], reservation_ID:[4], customer:[Katq Jeleva], date:[2021-02-01], period:[7], room number:[202]
employee:[Roza Ivanova], reservation_ID:[6], customer:[Kristiqn Tomov], date:[2021-03-03], period:[5], room number:[403]
employee:[Roza Ivanova], reservation_ID:[15], customer:[Galin Plamenov], date:[2021-08-02], period:[14], room number:[401]
employee:[Roza Ivanova], reservation_ID:[19], customer:[Malina Todorova], date:[2021-10-25], period:[7], room number:[705]


PL/SQL procedure successfully completed.
```