Triggers

Триггер который меняет статус(color) в зависимости успеха операции

create or replace TRIGGER update\_patient\_color

AFTER INSERT ON surgery

FOR EACH ROW

BEGIN

IF :NEW.surgery\_success = 'true' THEN

UPDATE patient SET color = 'yellow' WHERE p\_id = :NEW.p\_id;

ELSE

UPDATE patient SET color = 'black' WHERE p\_id = :NEW.p\_id;

END IF;

END;

Триггер который распределяет в морг пациентов которые имеют статус(color) = “Black(dead)”

create or replace TRIGGER transfer\_to\_morgue

AFTER UPDATE ON patient

FOR EACH ROW

BEGIN

IF :NEW.color = 'black' THEN

INSERT INTO morgue (p\_id, death\_date)

SELECT :NEW.p\_id, inpatient.treatment\_time

FROM inpatient

WHERE inpatient.p\_id = :NEW.p\_id;

END IF;

END;

Удаляем из базы уже оплаченные счета

create or replace TRIGGER BillRemove

BEFORE DELETE ON Bill

FOR EACH ROW

BEGIN

IF :OLD.Bill\_paid = 'true' THEN

DELETE FROM Bill WHERE Bill\_id = :OLD.Bill\_id;

END IF;

END;

Триггер который отправляет в палату пациентов с жёлтым статусом

create or replace TRIGGER add\_yellow\_patient

BEFORE INSERT ON patient

FOR EACH ROW

BEGIN

-- check if the patient's color is yellow

IF :new.color = 'yellow' THEN

SELECT room.room\_no INTO patient.room\_no

FROM room r

WHERE (SELECT COUNT(p.p\_id) FROM patients p WHERE p.room\_no = r.room\_no) < r.beds\_no

FETCH FIRST ROW ONLY;

END IF;

END;

VIEWS

View который показывает ночных работников Госпиталя

CREATE OR REPLACE FORCE EDITIONABLE VIEW Nigth\_workers

("Staff\_id", "Name", "Type","Workshift")

AS SELECT Staff\_id,Name,type,Workshift

FROM Staff WHERE workshift = '20:00- 8:00' order by type;

View который показывает самый большой чек который нужно заплатить

CREATE OR REPLACE FORCE EDITIONABLE VIEW Biggest\_bill

("BILL\_ID", "ADRESAT\_id", "Bill\_purpose", "Bill\_total", "Bill\_paid") AS SELECT Bill\_id,Addresat\_id,Bill\_purpose,Bill\_total,Bill\_paid

FROM Bill WHERE bill\_purpose = 'Bill'

ORDER BY Bill\_total DESC

View который показывает зарплаты работников по убыванию

CREATE OR REPLACE FORCE EDITIONABLE VIEW STAFFS\_SALARY

("STAFF\_ID", "NAME", "AGE", "EXPERIENCE", "TYPE", "WORKSHIFT", "SALARY")

AS SELECT STAFF\_ID,NAME,AGE,EXPERIENCE,TYPE,WORKSHIFT,SALARY

FROM Staff

ORDER BY Salary DESC

**AGGREGATIONS**

Вывод среднего опыта сотрудников

SELECT AVG(EXPERIENCE)

FROM STAFF

Вывод средней зарплаты, сумарного бюджета, и количества всех сотрудников

SELECT AVG (SALARY), SUM (SALARY), COUNT(SALARY)

FROM STAFF;

Вывод максимального счета и минимального

select max(bill\_total),min(bill\_total)

from bill

Transactions

INSERT INTO PATIENT,INPATIENT если не p\_id будет занять то выводить ошибку

BEGIN

INSERT INTO patient (p\_id, name, color,age,address, sex)

VALUES (:p\_id, :name, :color, :age, :address, :sex);

INSERT INTO inpatient(p\_id,name ,department\_name ,Room\_no,treatment\_type,treatment\_time)

values (:p\_id,:name ,:department\_name ,:Room\_no,:treatment\_type,:treatment\_time);

EXCEPTION

WHEN dup\_val\_on\_index THEN

raise\_application\_error (-20001, 'Error: patient already exists');

END;

INSERT INTO MEDICINE,BILL если mdcn\_id or bill\_id будет занять то выводить ошибку

BEGIN

INSERT INTO medicine (mdcn\_id, mdcn\_name, mdcn\_cost,Company,mdcn\_date)

VALUES (:mdcn\_id, :mdcn\_name, :mdcn\_cost, :Company, :mdcn\_date);

INSERT INTO Bill(Bill\_id,Addresat\_id ,Bill\_purpose ,Bill\_total,bill\_paid)

values (:bill\_id,:adressat\_id ,:bil\_purpose ,:bill\_total,:bill\_paid);

EXCEPTION

WHEN dup\_val\_on\_index THEN

raise\_application\_error (-20001, 'Error: this medicine\_id already exist');

END;

UPDATE TABLE STAFF будет увеличивать зарплату каждый год

BEGIN

DBMS\_SCHEDULER.CREATE\_JOB(

job\_name => 'update\_staff\_salary',

job\_type => 'PLSQL\_BLOCK',

job\_action => 'BEGIN

UPDATE staff

SET experience = DATEADD(YEAR, 1, experience) and salary = 1000 \* (1 + experience/100) where type = ''Doctor'';

UPDATE staff

SET experience = DATEADD(YEAR, 1, experience) and salary = 300 \* (1 + experience/100) where type = ''Nurse'';

UPDATE staff

SET experience = DATEADD(YEAR, 1, experience) and salary = 250 \* (1 + experience/100) where type = ''Driver'';

UPDATE staff

SET experience = DATEADD(YEAR, 1, experience) and salary = 200 \* (1 + experience/100) where type = ''Technical\_staff'';

COMMIT;

END;',

start\_date => SYSDATE,

repeat\_interval => 'FREQ=YEARLY;BYMONTH=1;BYDAY=WE',

enabled => TRUE

);

END;

Subqueries

Выводить Id и Имя пациента у которого анализ провалилься

SELECT P\_ID, NAME

FROM patient

WHERE p\_id IN (SELECT P\_ID

FROM LABORATORY

WHERE ANALYZE\_TEST = 'Fail')

Выводить id пациента у которого комната под номером 240

SELECT \* FROM PATIENT

WHERE p\_id IN (SELECT p\_id

FROM INPATIENT

WHERE Room\_no = 240)

Выводить id пациента и время операций у кого вид лечения равен I69149

SELECT p\_id, Surgery\_date

FROM surgery

WHERE p\_id IN (SELECT p\_id

FROM inpatient

WHERE treatment\_type = 'I69149')

Indexes

Индекс у которого данные врачей с их профессиями,именами и номерами

CREATE INDEX DOCTOR\_TYPE\_CONNECTION

ON DOCTOR (doc\_type, name, phone\_number);

Индекс который показывает информация про операцию

CREATE INDEX SURGERY\_INFO

ON SURGERY(P\_ID,SURGERY\_DATE,SURGERY\_SUCCESS);

Индекс которых показывает успех/провал исследований

CREATE INDEX Research\_test

ON Laboratory(Research\_type,Analyze\_test);

JOIN

Показывает информация связанную с пациентом и его информаций в болнице

SELECT \* FROM Patient inner join inpatient on patient.p\_id = inpatient.p\_id