МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КПІ»

ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ

КАФЕДРА АСОІУ

Звіт

з дисципліни

«Організація баз даних та знань»

Лабораторна робота №6

Виконав :

Студент групи ІС-32

№Залікової книжки - 3407

Касянчик Д.О.

Перевірила :

Томахіна О. С.

Київ – 2015

1. **Мета роботи**

Опанування запитів, для роботи з послідовностями, тригерами, та тригерами.

1. **Опис предметної області**

База конструкторсько-технологічної інформації підприємства зберігає наступну + ієрархічній специфікації (Bill of Material, BOM). Складовими BOM є деталі або зборочні одиниці (ДЗО). Спеціфікація продукту, як і специфікації його складових (зборочних одиниць, тобто агрегатів), містить перелік деталей або зборочних одиниць (ДЗО), які входять у даний продукт або агрегат. ДЗО або закупаються, або виробляються. Кожна ДЗО має вагу і собівартість.

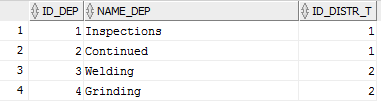
Якщо ДЗО виробляється на підприємстві, Відділ Головного технолога визначає для неї технологічний маршрут по певних дільницях певних цехів, починаючи з тої дільниці, де вона виникає вперше, і закінчуючи операцією зборки, згідно якій вона потрапляє в зборочну одиницю вищого рівня.

На кожній дільниці над ДЗО, яка там обробляється, виконуються певні технологічні операції певної тривалості, яка залежить від ДЗО. Кожна технологічна операція має вартість одиниці часу (хвилини), куди входять вартість часу роботи обладнання і вартість часу роботи робітника, що обслуговує це обладнання.

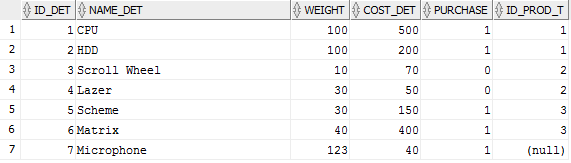
По цій базі менеджери підприємства аналізують планову собівартість виробництва продуктів.

1. **Таблиці використанні в вправах**

DEPARTMENT



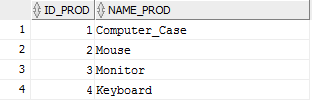
DETAILS



DISTRICT

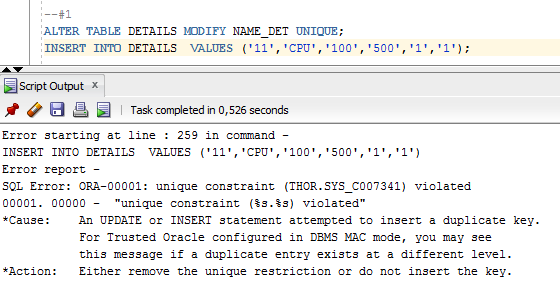


PRODUCT



1. **Вправи**
2. ***Налаштування унікальності поля. Зробити обмеження на поля :***
3. ***обов’язковості заповнення.***
4. ***унікальності значення поля.***
5. ***первинного ключа***
6. ***зовнішнього ключа .***
7. ***перевірки на значення.***

Так, як ці обмеження були використані в першій лабараторній роботі, я наведу лише ті, які були задіяні вперше в цій лабараторній роботі.



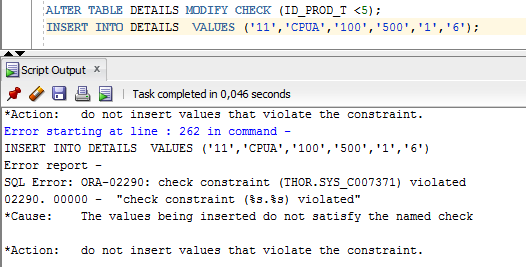
Screen Shot “Unique”

Код :

ALTER TABLE DETAILS MODIFY NAME\_DET UNIQUE;

INSERT INTO DETAILS VALUES ('11','CPU','100','500','1','1');

Як ми бачимо по Screen Shot’i було виведено, що ми не можемо вставити цей рядок, так, як значення NAME\_DET має бути унікальним.



Screen Shot “CHECK”

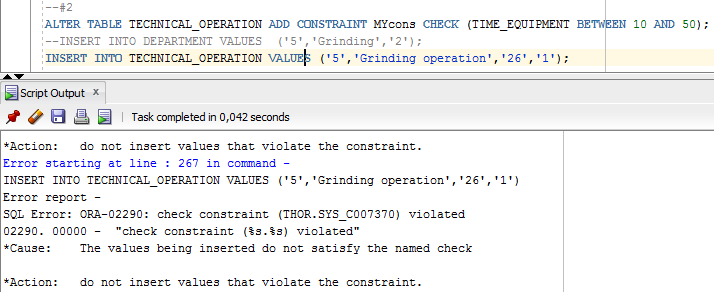
Код :

ALTER TABLE DETAILS MODIFY CHECK (ID\_PROD\_T <5);

INSERT INTO DETAILS VALUES ('11','CPUA','100','500','1','6');

Як ми бачимо по Screen Shot’i було виведено, що ми не можемо вставити цей рядок, так, як значення, не задовольняє обмеженню поля.

1. ***Налаштування обмежень рівня атрибутів або кортежів***

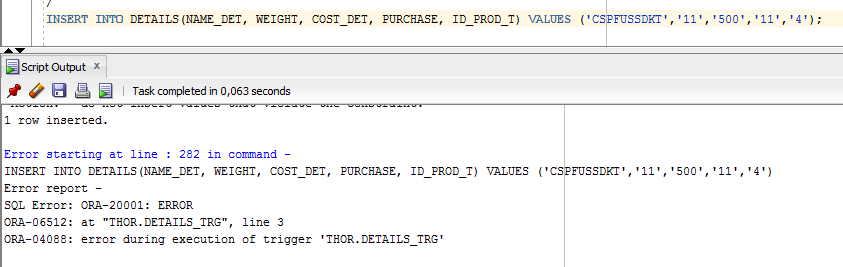


КОД :

ALTER TABLE TECHNICAL\_OPERATION ADD CONSTRAINT MYcons CHECK (TIME\_EQUIPMENT BETWEEN 10 AND 50);

Як ми бачимо по Screen Shot’i було виведено помилку, так, як ми вставляємо значення, яке не задовольняє обмеженню поля.

1. ***Тригери***



КОД :

CREATE OR REPLACE TRIGGER DETAILS\_TRG

BEFORE INSERT ON DETAILS

FOR EACH ROW

BEGIN

IF NOT(:NEW.WEIGHT > 20) THEN

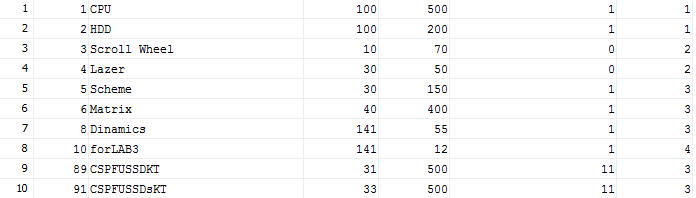
raise\_application\_error(-20001,'ERROR');

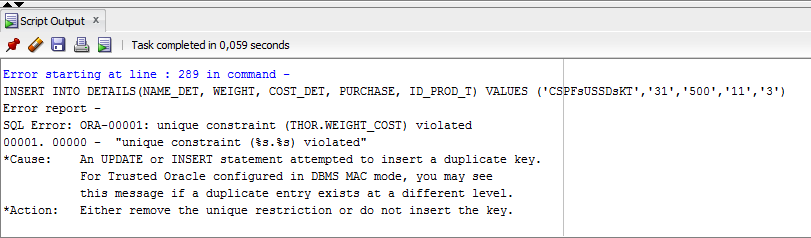
END IF;

END;

Як ми бачимо по Screen Shot’i був створений тригер, який накладав обмеження на поле «Вага», тобто при вставці якогось нового рядка, в якому поле «Вага» не задовольняло умові, що воно має бути більше 20 , тригер не вставляв даний рядок.

1. ***Індекси***





КОД :

CREATE INDEX NameIndex ON DEPARTMENT(NAME\_DEP);

CREATE UNIQUE INDEX WEIGHT\_COST ON DETAILS(COST\_DET, WEIGHT);

Як ми бачимо по Screen Shot’i була створені INDEX’и. І оскільки ми створили індекс по двох значеннях COST\_DET і WEIGHT, то ми не можемо додавати рядки з значеннями COST\_DET і WEIGHT які дорівнюють відповідним значенням в будь-якому іншому рядку.

1. ***Послідовності***

Код :

ALTER SEQUENCE WORK\_TRIGGER\_SEQ

INCREMENT BY 1

MAXVALUE 1000

MINVALUE 0

NOCYCLE

NOCACHE;

Код тригера :

CREATE OR REPLACE TRIGGER WORK\_TRIGGER\_\_TRG

BEFORE INSERT ON DETAILS

FOR EACH ROW

BEGIN

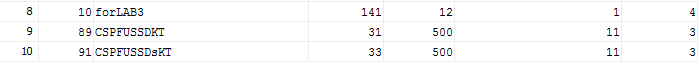
SELECT WORK\_TRIGGER\_SEQ.nextval INTO :new.ID\_DET FROM dual;

END;

/

Дана послідовність була створена, для того, щоб працював автоінкремент в таблиці DETAILS. Для того, щоб працював автоінкремент, ми також створили тригер для цієї таблиці, який перед кожною вставкою рядка в цю таблицю, визначав значення ID, як значення з послідовності.

Screen Shot “DETAILS” з використанням тригера



1. **Висновок**

В процесі виконання даної лабараторної роботи ознайомився з запитами мови SQL. Ознайомився з новими обмеженнями, а саме UNIQUE та CHECK. Навчився основам роботи з тригерами та послідовностями. Дізнався та зрозумів принцип роботи індексів в SQL.

1. **Код реалізації БД**

CREATE TABLE PRODUCT

(

ID\_PROD INT PRIMARY KEY ,

NAME\_PROD CHAR (30) NOT NULL

--FOREIGN KEY (ID\_PROD) REFERENCES DETAILS (ID\_PROD\_T)

);

CREATE TABLE DETAILS

(

ID\_DET INT PRIMARY KEY,

NAME\_DET CHAR (30) NOT NULL,

WEIGHT INT NOT NULL,

COST\_DET INT NOT NULL,

PURCHASE INT NOT NULL ,

ID\_PROD\_T INT ,

FOREIGN KEY (ID\_PROD\_T) REFERENCES PRODUCT (ID\_PROD)

);

CREATE TABLE DISTRICT

(

ID\_DISTR INT PRIMARY KEY,

NAME\_DISTR CHAR (30) NOT NULL

--FOREIGN KEY (ID\_DET\_T) REFERENCES DETAILS (ID\_DET)

);

CREATE TABLE DISTRICT\_DETAILS

(

ID\_DET\_T INT ,

ID\_DISTR\_T INT,

FOREIGN KEY (ID\_DET\_T) REFERENCES DETAILS (ID\_DET),

FOREIGN KEY (ID\_DISTR\_T) REFERENCES DISTRICT (ID\_DISTR)

);

CREATE TABLE DEPARTMENT

(

ID\_DEP INT PRIMARY KEY,

NAME\_DEP CHAR (30) NOT NULL,

ID\_DISTR\_T INT NOT NULL,

FOREIGN KEY (ID\_DISTR\_T) REFERENCES DISTRICT (ID\_DISTR)

);

CREATE TABLE TECHNICAL\_OPERATION

(

ID\_TECHN INT PRIMARY KEY,

NAME\_TECHN CHAR(30) NOT NULL,

TIME\_EQUIPMENT INT ,

TIME\_WORKER INT ,

--D\_DEP\_T INT NOT NULL,

--ID\_DET\_TECH INT ,

FOREIGN KEY (ID\_TECHN) REFERENCES DEPARTMENT (ID\_DEP)

--FOREIGN KEY (ID\_DET\_TECH) REFERENCES DETAILS (ID\_DET)

);

CREATE TABLE OPERATION\_DETAILS

(

ID\_DET\_TECH INT ,

ID\_TECHN\_T INT,

FOREIGN KEY (ID\_DET\_TECH) REFERENCES DETAILS (ID\_DET),

FOREIGN KEY (ID\_TECHN\_T) REFERENCES TECHNICAL\_OPERATION (ID\_TECHN)

);

--ALTER TABLE OPERATION\_DETAILS MODIFY ID\_DET\_TECH int not null;

ALTER TABLE OPERATION\_DETAILS MODIFY ID\_TECHN\_T INT NOT NULL;

ALTER TABLE OPERATION\_DETAILS MODIFY ID\_DET\_TECH INT NOT NULL;

ALTER TABLE DETAILS MODIFY COST\_DET INT NULL;

INSERT INTO PRODUCT VALUES ('1','Computer\_Case');

INSERT INTO PRODUCT VALUES ('2','Mouse');

INSERT INTO PRODUCT VALUES ('3','Monitor');

INSERT INTO PRODUCT VALUES ('4','Keyboard');

INSERT INTO DETAILS VALUES ('1','CPU','100','500','1','1');

INSERT INTO DETAILS VALUES ('2','HDD','100','200','1','1');

INSERT INTO DETAILS VALUES ('3','Scroll Wheel','10','70','0','2');

INSERT INTO DETAILS VALUES ('4','Lazer','30','50','0','2');

INSERT INTO DETAILS VALUES ('5','Scheme','30','150','1','3');

INSERT INTO DETAILS VALUES ('6','Matrix','40','400','1','3');

INSERT INTO DETAILS VALUES ('7','Microphone','123','40','1','');

INSERT INTO DETAILS VALUES ('8','Dinamics','141','','1','');

INSERT INTO DISTRICT VALUES ('1','Reability of Resourse');

INSERT INTO DISTRICT VALUES ('2','Creation Details');

INSERT INTO DISTRICT\_DETAILS VALUES ('2','1');

INSERT INTO DISTRICT\_DETAILS VALUES ('6','1');

INSERT INTO DISTRICT\_DETAILS VALUES ('2','2');

INSERT INTO DISTRICT\_DETAILS VALUES ('5','2');

INSERT INTO DEPARTMENT VALUES ('1','Inspections','1');

INSERT INTO DEPARTMENT VALUES ('2','Continued','1');

INSERT INTO DEPARTMENT VALUES ('3','Welding','2');

INSERT INTO DEPARTMENT VALUES ('4','Grinding','2');

INSERT INTO TECHNICAL\_OPERATION VALUES ('1','Testing operation','23','12');

INSERT INTO TECHNICAL\_OPERATION VALUES ('2','Repairing opertion','32','12');

INSERT INTO TECHNICAL\_OPERATION VALUES ('3','Welding operation','41','46');

INSERT INTO TECHNICAL\_OPERATION VALUES ('4','Grinding operation','26','28');

INSERT INTO OPERATION\_DETAILS VALUES ('2','2');

INSERT INTO OPERATION\_DETAILS VALUES ('6','1');

INSERT INTO OPERATION\_DETAILS VALUES ('2','4');

INSERT INTO OPERATION\_DETAILS VALUES ('5','2');

INSERT INTO OPERATION\_DETAILS VALUES ('1','1');

INSERT INTO OPERATION\_DETAILS VALUES ('8','2');

-------------------------LAB 3

--#1

SELECT \* FROM DETAILS ORDER BY COST\_DET;

--#2

SELECT \* FROM DETAILS WHERE COST\_DET > 200 AND PURCHASE = 1 AND WEIGHT >50;

--#3

SELECT \* FROM PRODUCT WHERE EXISTS (SELECT \* FROM DETAILS WHERE ID\_PROD\_T >= '2' AND ID\_PROD = ID\_PROD\_T );

SELECT \* FROM DETAILS WHERE ID\_DET IN (SELECT ID\_DET\_TECH FROM OPERATION\_DETAILS);

--#4

SELECT \* FROM DETAILS

INNER JOIN PRODUCT

ON DETAILS.ID\_PROD\_T =PRODUCT.ID\_PROD;

SELECT \* FROM PRODUCT

LEFT JOIN DETAILS

ON DETAILS.ID\_PROD\_T =PRODUCT.ID\_PROD;

SELECT \* FROM PRODUCT

RIGHT JOIN DETAILS

ON DETAILS.ID\_PROD\_T =PRODUCT.ID\_PROD;

--#5

SELECT NAME\_DEP FROM DEPARTMENT

UNION

SELECT NAME\_DISTR FROM DISTRICT;

--#6

SELECT CAST(WEIGHT AS varchar(4)) from DETAILS;

SELECT NAME\_DET FROM DETAILS

WHERE NAME\_DET LIKE '%er%'

OR NAME\_DET LIKE 'Sc%';

UPDATE DETAILS SET COST\_DET = '0' WHERE COST\_DET IS NULL;

-------------------------LAB 4

--#1

SELECT ID\_PROD\_T, SUM(COST\_DET) AS "TOTAL COST"

FROM DETAILS

GROUP BY ID\_PROD\_T

ORDER BY "TOTAL COST";

--#2

SELECT ID\_PROD\_T, SUM(COST\_DET) AS "TOTAL COST"

FROM DETAILS

GROUP BY ID\_PROD\_T

HAVING SUM(COST\_DET) > 500 ;

--#3

CREATE TABLE FACTORY AS

(

SELECT ID\_DEP, NAME\_DEP, NAME\_DISTR

FROM DISTRICT, DEPARTMENT WHERE DEPARTMENT.ID\_DISTR\_T = DISTRICT.ID\_DISTR

);

--#4

INSERT INTO DETAILS VALUES ('9','TouchPad','11','','11','');

UPDATE DETAILS

SET COST\_DET = dbms\_random.value(20, 213)

WHERE COST\_DET IS NULL;

--#5

INSERT INTO DETAILS VALUES ('11','TouchPad\_FOR\_DEL','11111','','11','');

DELETE FROM DETAILS

WHERE WEIGHT >= 10000;

--#6

ALTER TABLE FACTORY MODIFY NAME\_DISTR VARCHAR(100) NULL;

INSERT INTO FACTORY (ID\_DEP ,NAME\_DEP )

SELECT ID\_TECHN+4 ,NAME\_TECHN FROM TECHNICAL\_OPERATION WHERE TIME\_EQUIPMENT <= 30;

--# захист лаби 3

INSERT INTO DETAILS VALUES ('10','forLAB3','141','12','1','4');

INSERT INTO DISTRICT\_DETAILS VALUES ('10','1');

SELECT \* FROM DISTRICT

WHERE ID\_DISTR IN

(

SELECT DISTRICT\_DETAILS.ID\_DISTR\_T FROM DISTRICT\_DETAILS

WHERE DISTRICT\_DETAILS.ID\_DET\_T IN

(

SELECT ID\_DET FROM DETAILS

WHERE ID\_PROD\_T = 4

)

)

;

SELECT \* FROM DISTRICT

RIGHT JOIN DISTRICT\_DETAILS ON ID\_DISTR = DISTRICT\_DETAILS.ID\_DISTR\_T

RIGHT JOIN DETAILS ON DISTRICT\_DETAILS.ID\_DET\_T = ID\_DET

WHERE ID\_PROD\_T = 4

;

--# захист лаби 4

SELECT \* FROM TECHNICAL\_OPERATION

WHERE ID\_TECHN IN

(

SELECT ID\_TECHN\_T FROM OPERATION\_DETAILS

WHERE ID\_DET\_TECH IN

(

SELECT ID\_DET FROM DETAILS D1

WHERE EXISTS(

SELECT SUM(COST\_DET) FROM DETAILS D2

WHERE D1.ID\_PROD\_T =D2.ID\_PROD\_T

GROUP BY ID\_PROD\_T

HAVING SUM(COST\_DET) =

(

SELECT MAX (SUM(COST\_DET)) FROM DETAILS

GROUP BY ID\_PROD\_T

)

OR SUM(COST\_DET) =

(

SELECT MIN (SUM(COST\_DET)) FROM DETAILS

GROUP BY ID\_PROD\_T

)

)

)

);

SELECT ID\_DET FROM DETAILS D1

WHERE EXISTS(

SELECT SUM(COST\_DET) FROM DETAILS D2

WHERE D1.ID\_PROD\_T =D2.ID\_PROD\_T

GROUP BY ID\_PROD\_T

HAVING

SUM(COST\_DET) =

(

SELECT MAX (SUM(COST\_DET)) FROM DETAILS

GROUP BY ID\_PROD\_T

)

OR

SUM(COST\_DET) =

(

SELECT MIN (SUM(COST\_DET)) FROM DETAILS

GROUP BY ID\_PROD\_T

)

);

-------------------------LAB 6

--#1

ALTER TABLE DETAILS MODIFY NAME\_DET UNIQUE;

INSERT INTO DETAILS VALUES ('11','CPU','100','500','1','1');

ALTER TABLE DETAILS MODIFY CHECK (ID\_PROD\_T <5);

INSERT INTO DETAILS VALUES ('11','CPUA','100','500','1','6');

--#2

ALTER TABLE TECHNICAL\_OPERATION ADD CONSTRAINT MYcons CHECK (TIME\_EQUIPMENT BETWEEN 10 AND 50);

--INSERT INTO DEPARTMENT VALUES ('5','Grinding','2');

INSERT INTO TECHNICAL\_OPERATION VALUES ('5','Grinding operation','26','1');

--#3

INSERT INTO DETAILS(NAME\_DET, WEIGHT, COST\_DET, PURCHASE, ID\_PROD\_T) VALUES ('CPFUKT','100','500','11','4');

--

CREATE OR REPLACE TRIGGER DETAILS\_TRG

BEFORE INSERT ON DETAILS

FOR EACH ROW

BEGIN

IF NOT(:NEW.WEIGHT > 20) THEN

raise\_application\_error(-20001,'ERROR');

END IF;

END;

/

INSERT INTO DETAILS(NAME\_DET, WEIGHT, COST\_DET, PURCHASE, ID\_PROD\_T) VALUES ('CSPFUSSDKT','11','500','11','4');

--#4

CREATE INDEX NameIndex ON DEPARTMENT(NAME\_DEP);

CREATE UNIQUE INDEX WEIGHT\_COST ON DETAILS(COST\_DET, WEIGHT);

INSERT INTO DETAILS(NAME\_DET, WEIGHT, COST\_DET, PURCHASE, ID\_PROD\_T) VALUES ('CSPFsUSSDsKT','31','500','11','3');

--#5

ALTER SEQUENCE WORK\_TRIGGER\_SEQ

INCREMENT BY 1

MAXVALUE 1000

MINVALUE 0

NOCYCLE

NOCACHE;

CREATE OR REPLACE TRIGGER WORK\_TRIGGER\_\_TRG

BEFORE INSERT ON DETAILS

FOR EACH ROW

BEGIN

SELECT WORK\_TRIGGER\_SEQ.nextval INTO :new.ID\_DET FROM dual;

END;

/

INSERT INTO DETAILS(NAME\_DET, WEIGHT, COST\_DET, PURCHASE, ID\_PROD\_T) VALUES ('SSKDHFDDOSSST','711','500','11','4');