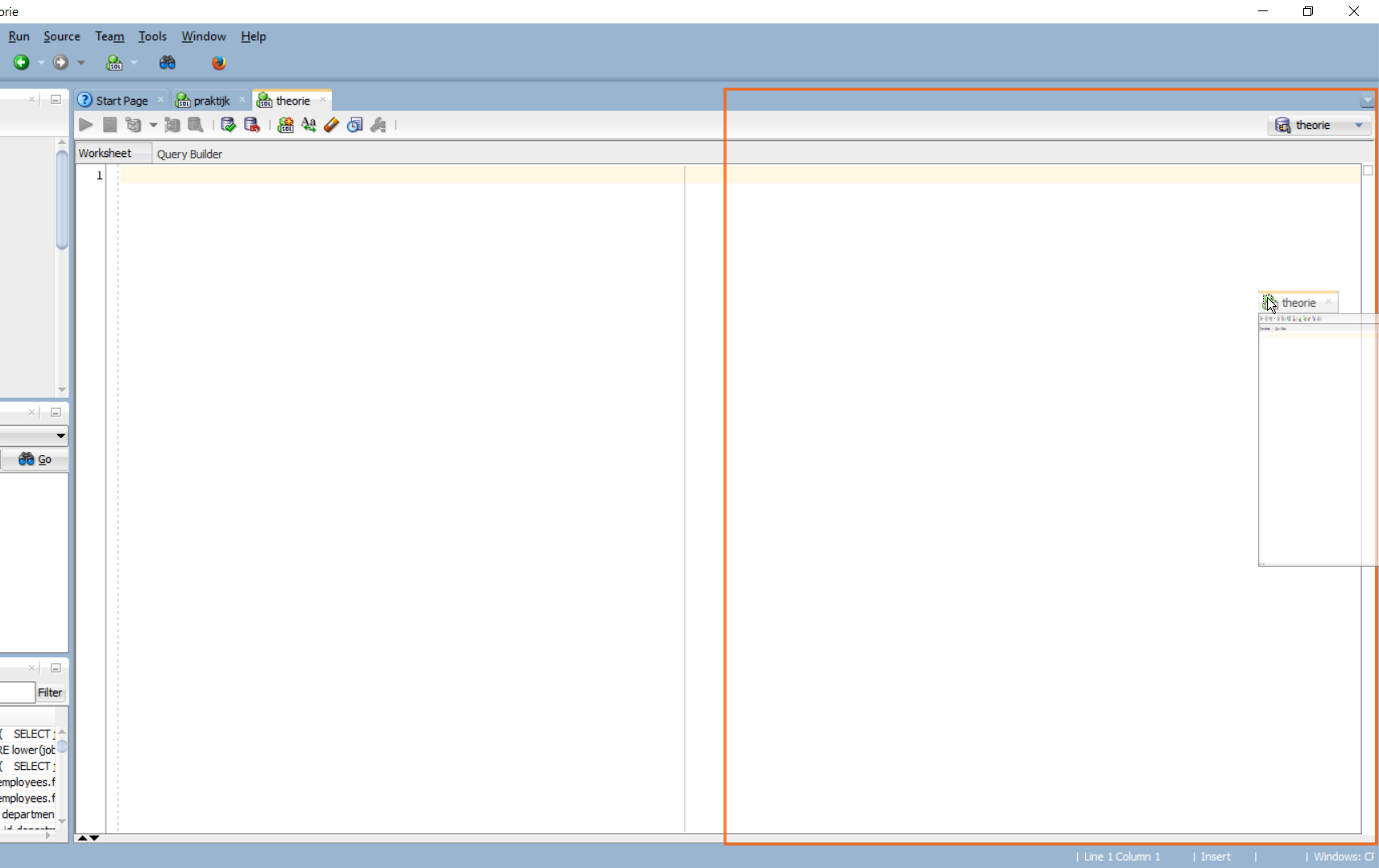
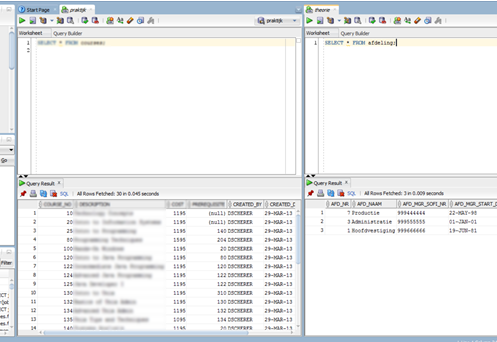
**PROJECT 5**

Transacties en indexen

Open in SQLDeveloper een SQL worksheet voor gebruiker THEORIE én voor gebruiker PRAKTIJK. Sleep te tab van één van de vensters naar rechts, zodat de vensters naast mekaar staan.





**In de praktijk worksheet:**

Geef gebruiker THEORIE de toelating om op de tabel personeel te werken:

*GRANT ALL ON personeel TO theorie;*

OPGELET: als de gebruiker THEORIE de tabel personeel van gebruiker PRAKTIJK aanspreekt, moet hij de **volledige** tabelnaam gebruiken: schema.tabelnaam DUS praktijk. personeel

1. **In de praktijk worksheet**

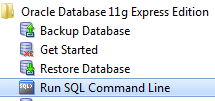
Verander de voornaam van het personeelslid met sofi\_nr 215243964 naar Magda. Bekijk het resultaat met een SQL commando.

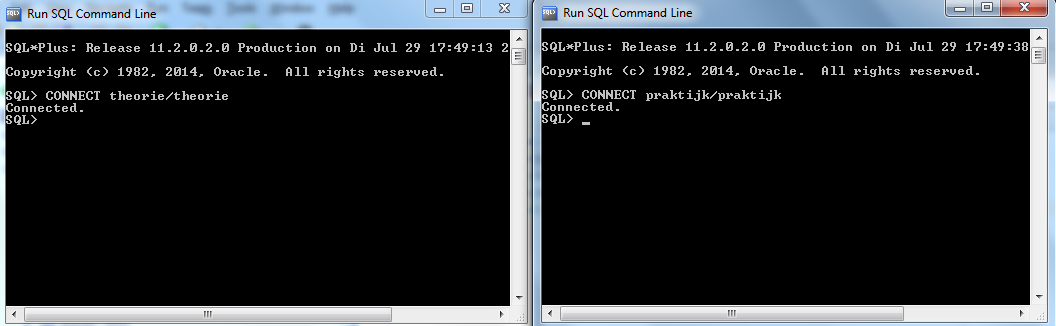
1. **In de theorie worksheet**

Bekijk de gegevens van de gebruiker met sofi\_nr . 215243964 met een SQL commando.

Verklaar het resultaat.

1. **Open via *Run SQL command line* een SQL\*PLUS sessie en verbind als de gebruiker praktijk.**

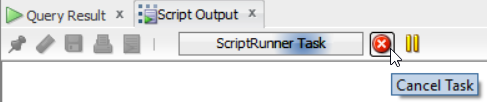




Bekijk de gegevens van de gebruiker met sofi\_nr . 215243964 met een SQL commando.

Verklaar het resultaat.

1. Klik rechts op Connections>praktijk en kies "Open SQL Worksheet". Je opent een EXTRA worksheet voor gebruiker praktijk. Bekijk in de extra worksheet (praktijk~1) het personeelslid met sofi\_nr . 215243964.
2. Schrijf de veranderingen definitief naar de databank in de EXTRA PRAKTIJK SQL worksheet (met venstertab praktijk~1). Bekijk de gegevens van het personeelslid met sofi\_nr . 215243964 via de andere worksheets en via de SQL\*PLUS commandolijn. Verklaar het resultaat.
3. Sluit de SQL\*PLUS sessie en de extra praktijk SQL worksheet (praktijk~1). Voor zowel theorie en praktijk staat dus nog één worksheet open.
4. **In de theorie worksheet:**  wijzig de voornaam van het personeelslid met sofi\_nr . 215243964 naar Maria Magdalena (in twee woorden).
5. In deze stap voer je enkele aanpassingen uit op de databank. Verklaar voor elke aanpassing het gedrag van de databank. Indien één van de instructies in deze stap hangt, annuleer dan door op de rode knop met X in het resultaatpaneel te drukken.



**In de praktijk worksheet:**

* 1. Wijzig de voornaam van het personeelslid met sofi\_nr . 215243964 naar Maria
  2. Wijzig het tussenvoegsel van het personeelslid met sofi\_nr . 215243964 naar "van der"
  3. Wijzig de voornaam van het personeelslid met sofi\_nr . 216223308 naar Maria

1. In de theorie worksheet: De tussenvoegsels in de databank moeten met een hoofdletter geschreven worden. Voor deze verandering door in de databank voor alle personeelsleden die niet het tussenvoegsel 'de' hebben (zij zijn van adel).

**In de praktijk worksheet:** Voer volgende stappen uit en verklaar hete gedrag:

1. Achter afkortingen van academische titels hoort een punt. Selecteer en vergrendel alle personeelsleden met ziekenhuistitel "dr". Wacht niet langer dan 5 seconden op het resultaat.
2. Om werken met namen makkelijker te maken verander je alle tussenvoegsels die NULL zijn door een lege String.
3. Maak de aanpassingen van gebruiker theorie in de praktijk databank ongedaan. Bepaal zelf in welk venster je dit best doet.

Alle verdere oefeningen verlopen in de praktijk worksheet:

1. Voeg toe dat personeelslid met persnr 66444 (Magda Lanoote) vandaag de specialisatie VPK3 behaalde.

Verifieer dat de specialisatie in de databank aanwezig is.

1. Voeg een intermediair herstelpunt toe.
2. Schrap de oude specialisatie (VPK1) van deze verpleegster met de instructie

DELETE FROM pers\_specialisaties ; WHERE persnr=66444 AND code = 'VPK1';

1. Er zat een ';' teveel in de vorige instructie, waardoor **alle** specialisatie geschrapt zijn. Maak deze laatste aanpassing ongedaan. Verifieer het resultaat.
2. Verwijder de specialisatie 'VPK1' voor de verlpleegster nu op correcte wijze.
3. Maak de verwijdering van VPK1 terug ongedaan. Kan je meerdere keren naar hetzelfde savepoint teruggaan? Verifieer of er nu terug twee specialisaties zijn voor Magda Lanoote.
4. De doogegeven specialisatie was niet correct. Verander de specialisatie VPK3 van magda Lanoote naar VPK2.
5. Schap alle data uit de tabel patientfiches met de instructie

TRUNCATE TABLE patientfiches;

1. Annuleer de acties, tot net na de invoering van specialisatie VPK3 voor Magda Lanoote. Leg uit wat er gebeurt.
2. :

Zoek in de dictionary tabellen welke indexen er momenteel op de tabellen uit de praktijk database staan. Wanneer werden deze aangemaakt? Wat merk je aan de naamgeving?

1. :

Creëer op de attributen OPNAMENR en PERSNR uit de tabel BEHANDELINGEN een samengestelde index. In welke volgorde plaats je de attributen en waarom?

1. :
2. Voer het volgende script uit:

ALTER SESSION SET OPTIMIZER\_MODE=FIRST\_ROWS;

exec dbms\_stats.gather\_schema\_stats (NULL,dbms\_stats.auto\_sample\_size);

Creëer een index ind\_achternaam\_ptnt op het attribuut ACHTERNAAM uit de tabel PATIENTEN.

1. Voer de volgende select uit:

SELECT \* FROM patienten

WHERE achternaam = 'Overdijk';

Bekijk via de EXPLAIN PLAN knop boven de worksheet (of F10) of de zojuist gecreëerde index bij de uitvoering van de instructie werd gebruikt.

1. Voer de volgende select uit en bekijk opnieuw EXPLAIN PLAN. Is er een verschil met de vorige uitvoering? Leg uit.

SELECT \* FROM patienten

WHERE lower(achternaam) = 'Overdijk';

1. Maak een function based index om bovenstaand probleem op te lossen.

*CREATE INDEX ind\_lower\_achternaam\_ptnt*

*ON patienten(lower(achternaam));*

Voer de query opnieuw uit en controleer via het EXPLAIN PLAN of de function based index werd gebruikt.

1. Selecteer alle klanten waarvan de achternaam met Ov begint. Bekijk het EXPLAIN PLAN en leg uit.
2. Selecteer alle klanten waarvan de achternaam "man" bevat. Bekijk het EXPLAIN PLAN en leg uit.
3. Ga na welke indexen op de tabel PATIENTEN staan.
4. Probeer elke index te verwijderen. Wat is het resultaat? Leg uit.
5. => optioneel

Zoek op in de data dictionary:

1. Welke tabellen zijn gedefinieerd in het huidige schema. Vraag uit de dictionary tabel USER\_TABLES de volgende informatie op: tabelnamen en de naam van de tablespace waarin de tabellen zitten.
2. Welke constraints staan er op de tabel BEHANDELINGEN? Vraag tabelnaam, constraintnaam, constrainttype en zoekvoorwaarde.
3. Geef voor alle CHECK constraints op de tabellen BEHANDELINGEN en VERRICHTINGEN: naam, zoekvoorwaarde en de tabel waarop ze geldig zijn.

**Herstel de praktijk databank terug in haar oorspronkelijke staat door de maak en vul scripts opnieuw uit te voeren.**

**Oplossingen**



UPDATE personeel SET voornaam = 'Magda' WHERE sofi\_nr='215243964';

SELECT \* FROM personeel WHERE sofi\_nr='215243964';



SELECT \* FROM praktijk.personeel WHERE sofi\_nr='215243964';

Gebruiker praktijk heeft zijn aanpassing nog niet gecommit. READ CONSISTENCY is dus van toepassing.

Gebruiker theorie ziet de aanpassingen dus niet

SELECT \* FROM personeel WHERE sofi\_nr='215243964';

Het gaat om dezelfde gebruiker maar met een andere connectie. Ook hiervoor is READ CONSISTENCY van kracht. Je ziet de aangepaste data dus niet.

Je ziet de aanpassingen deze keer wel. De extra SQL worksheet gebruikt dezelfde connectie als de eerste worksheet.

COMMIT;

Je kan in de tweede praktijk worksheet de acties uit de eerste worksheets bevestigen. Beiden werken dus effectief in dezelfde transactie.

De commit heeft als resultaat dat alle connecties nu de aangepaste gegevens zien



UPDATE praktijk.personeel

SET voornaam = 'Maria Magdalena'

WHERE sofi\_nr='215243964';

UPDATE personeel SET voornaam = 'Maria' WHERE sofi\_nr='215243964';

-- I hangt: het veld is vergrendeld doordat gebruiker theorie dit veld aan het updaten is.

theorie dit veld aan het updaten is.

UPDATE reisburs

SET straat = 'AVENUE LOUISE', huisnr=59,postcode = 1050, gemeente = 'ELSENE'

WHERE bunr = 5;

-- II hangt: andere velden van dezelfde rij zijn ook vergrendeld

UPDATE personeel SET voornaam = Maria WHERE sofi\_nr='216223308';

-- III wordt uitgevoerd: andere rijen van dezelfde tabel zijn niet vergrendeld, zelfs niet als het om dezelfde kolom gaat.

UPDATE praktijk.personeel

SET tussenvoegsel = initcap(tussenvoegsel)

WHERE tussenvoegsel != 'de';

SELECT \*

FROM personeel

WHERE ziekenhuistitel='dr'

FOR UPDATE wait 5;

De select hangt want er zijn rijen met ziekenhuistitel 'dr' die een tussenvoegsel hebben end us geblokkeerd zijn door gebruiker theorie. Na 5 seconden krijg je de foutmelding:

ORA-30006: resource busy; acquire with WAIT timeout expired

update personeel

set tussenvoegsel=''

where tussenvoegsel is null;

Dit commando slaagt: de geblokeerde rijen (tussenvoegsel != 'de') zijn immers verschillend van de aangepaste rijen (tussenvoegsel is null).

1. In de theorie worksheet:

rollback;



INSERT INTO pers\_specialisaties VALUES (66444,'VPK3',sysdate);

SELECT \* FROM pers\_specialisaties WHERE persnr=66444 AND code = 'VPK3';

1. SAVEPOINT sauver\_svp;

ROLLBACK TO SAVEPOINT sauver\_svp;

SELECT \* FROM pers\_specialisaties

1. DELETE FROM pers\_specialisaties WHERE persnr=66444 AND code = 'VPK1';
2. ROLLBACK TO SAVEPOINT sauver\_svp;

Je kan meerdere rollbacks doen naar eenzelfde savepoint.

1. UPDATE pers\_specialisaties SET code='VPK2'

where code='VPK3' and persnr=66444;

2. ROLLBACK TO SAVEPOINT sauver\_svp;

SQL Error: ORA-01086: savepoint btl\_svp never established

TRUNCATE is een DDL-instructie met een impliciete COMMIT.

Je kan geen ROLLBACK uitvoeren en al je savepoints zijn verwijderd.

SELECT index\_name

FROM user\_indexes;

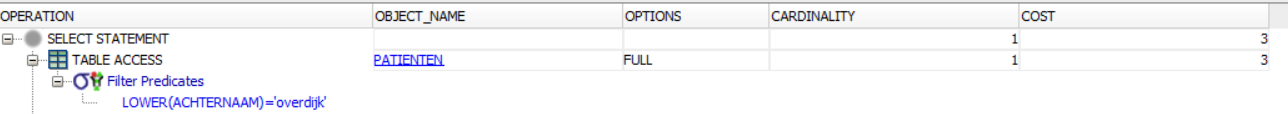
De indexen werden aangemaakt ten gevolge van de Primary Key constraints op de tabellen. Zij kregen ook de naam van de primary keys.

CREATE INDEX ind\_opnr\_persnr

on behandelingen(opnamenr, persnr);

Eerst op attribuut met meest verschillende waarden: oponamenr

1. CREATE INDEX ind\_achternaam\_ptnt ON patienten (achternaam);
2. De index op achternaam werd gebruikt.
3. Table access option full (Ful table scan)



De index wordt niet gebruikt wanneer het geïndexeerd attribuut onderworpen werd aan een functie.

1. Er wordt een index range scan gebruikt

SELECT \* FROM patienten WHERE achternaam LIKE 'Ov%';

Range scan: Hoewel er een LIKE operator werd gebruikt kan de volgorde in ind\_achternaam\_ptnt worden gebruikt.

SELECT \* FROM patienten WHERE lower(achternaam) LIKE '%man%';

De LIKE operator geeft resultaten die de volgorde van de index niet volgen. Oracle valt terug op een FULL TABLE SCAN.



SELECT index\_name,index\_type

FROM user\_indexes

WHERE table\_name = 'PATIENTEN';

INDEX\_NAME INDEX\_TYPE

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

IND\_LOWER\_ACHTERNAAM\_PTNT FUNCTION-BASED NORMAL

PK\_PATIENT NORMAL

U\_PAT\_SOFI\_NR NORMAL

IND\_ACHTERNAAM\_PTNT NORMAL



**DROP INDEX IND\_LOWER\_ACHTERNAAM\_PTNT;**

Index IND\_LOWER\_ACHTERNAAM\_PTNT dropped.

**DROP INDEX IND\_ACHTERNAAM\_PTNT;**

Index IND\_ACHTERNAAM\_PTNT dropped.

**DROP INDEX U\_PAT\_SOFI\_NR;**

ORA-02429: cannot drop index used for enforcement of unique/primary key

D**ROP INDEX PK\_PATIENT;**

ORA-02429: cannot drop index used for enforcement of unique/primary key

Je kan de unieke indexen enkel verwijderen door de unieke en key constraint te verwijderen.



**SELECT table\_name,tablespace\_name**

**FROM user\_tables;**



**SELECT table\_name,constraint\_name,constraint\_type, search\_condition**

**FROM user\_constraints**

**WHERE table\_name='BEHANDELINGEN';**

TABLE\_NAME CONSTRAINT\_NAME C SEARCH\_CONDITION

------------------------------ ------------------------------ - -----------------------BEHANDELINGEN NN\_BEHANDELING\_PATIENTNR C "OPNAMENR" IS NOT NULL

BEHANDELINGEN NN\_BEHANDELING\_PERSNR C "PERSNR" IS NOT NULL

BEHANDELINGEN NN\_BEHANDELING\_VERRICHTING\_NR C "VERRICHTINGNR" IS NOT NULL

BEHANDELINGEN CK\_BEH\_HUIDIGE\_REK\_TOTAAL C prijs\_beh >= 0

BEHANDELINGEN PK\_BEHANDELING P

BEHANDELINGEN FK\_BEHANDELING\_OPNAMEPATIENT R

BEHANDELINGEN FK\_BEHANDELING\_PERSONEEL R

BEHANDELINGEN FK\_BEHANDELING\_VERRICHTINGEN R



**SELECT table\_name,constraint\_name,search\_condition**

**FROM user\_constraints**

**WHERE constraint\_type='C'**

**AND table\_name IN ('BEHANDELINGEN','VERRICHTINGEN');**

TABLE\_NAME CONSTRAINT\_NAME SEARCH\_CONDITION

------------------------------ ------------------------------ -------------------------BEHANDELINGEN NN\_BEHANDELING\_PATIENTNR "OPNAMENR" IS NOT NULL

BEHANDELINGEN NN\_BEHANDELING\_PERSNR "PERSNR" IS NOT NULL

BEHANDELINGEN NN\_BEHANDELING\_VERRICHTING\_NR "VERRICHTINGNR" IS NOT NULL

BEHANDELINGEN CK\_BEH\_HUIDIGE\_REK\_TOTAAL prijs\_beh >= 0

VERRICHTINGEN NN\_VERRICHTING\_OMSCHRIJVING "OMSCHRIJVING" IS NOT NULL

VERRICHTINGEN CK\_VERRICHTING\_REK\_TOTAAL eenhpr\_verr >= 0