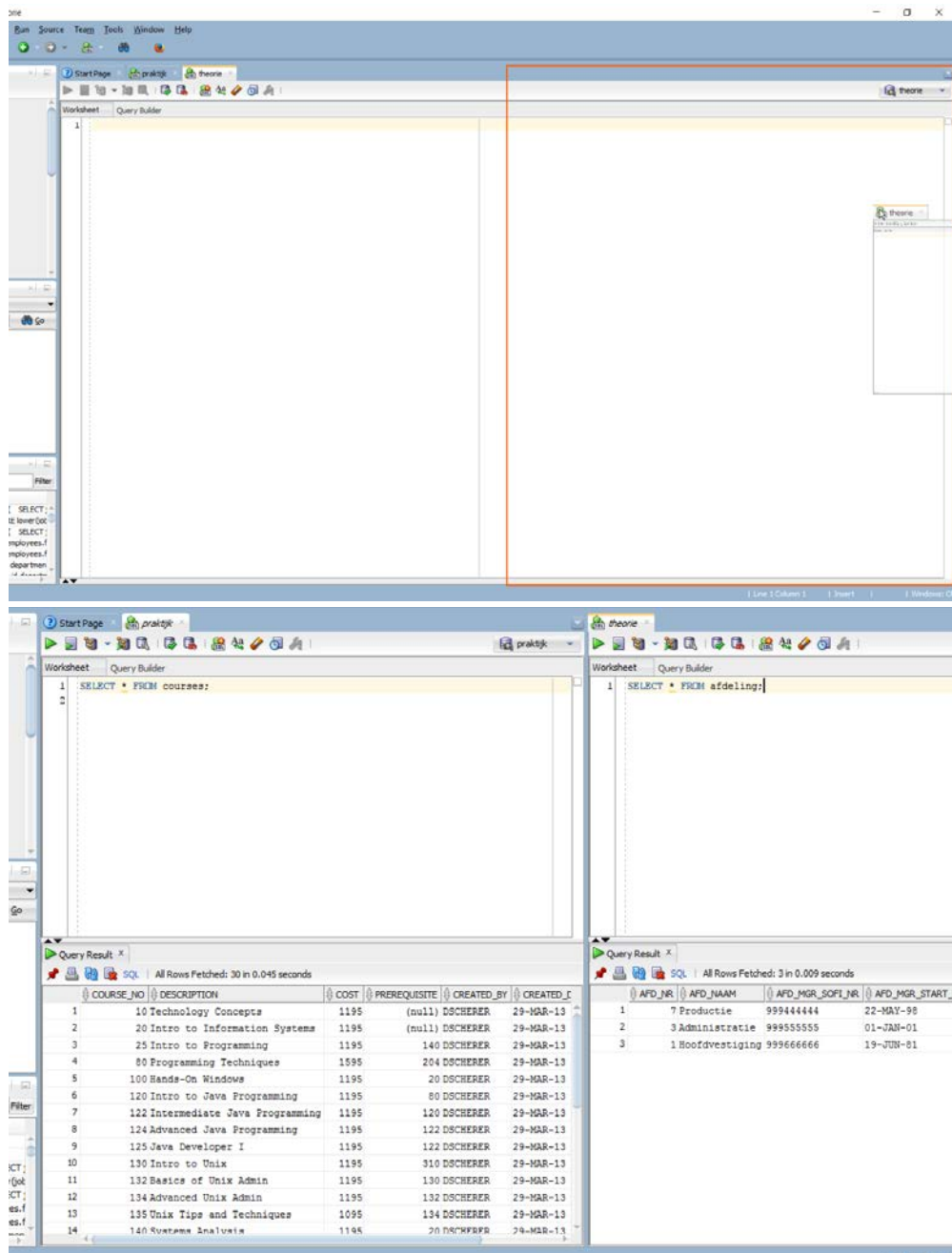


Oefeningen op selects en DML in een multi user omgeving

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



Laat gebruiker THEORIE rechten (SELECT en UPDATE) op de tabel MEDEWERKER geven aan gebruiker PRAKTIJK

GRANT SELECT, UPDATE ON medewerkers TO praktijk;

1.

Laat gebruiker THEORIE de onderstaande update uitvoeren:

```
UPDATE medewerkers  
SET plaats=UPPER(plaats)  
WHERE afd_nr=7;
```

Laat gebruiker PRAKTIJK selecteren op de tabel MEDEWERKER:

```
SELECT *  
FROM medewerkers;
```

Waarom foutmelding?

Tabel staat niet in zijn schema.

Hoe schrijf je de instructie correct?

```
SELECT *  
FROM theorie.medewerkers;
```

Ziet gebruiker PRAKTIJK de wijziging die door gebruiker THEORIE is doorgevoerd?

Neen

Hoe komt dat?

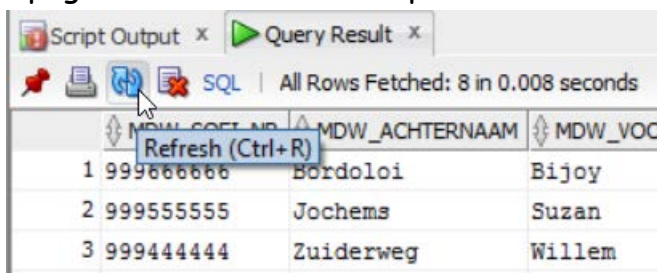
Is nog niet gecommitt !

Laat gebruiker THEORIE COMMIT geven

COMMIT;

Laat gebruiker PRAKTIJK opnieuw selecteren op de tabel MEDEWERKER.

Tip: gebruik de refresh knop:



Conclusie:

Nu, ziet praktijkgebruiker de wijzigingen WEL !

2.

a.

Laat gebruiker THEORIE onderstaande update uitvoeren:

```
UPDATE medewerkers  
SET salaris=salaris*1.05  
WHERE afd_nr=3;
```

Laat gebruiker PRAKTIJK onderstaande update uitvoeren:

```
UPDATE theorie.medewerkers  
SET salaris=salaris*1.05  
WHERE afd_nr=7;
```

Lukt dit? Waarom?

Ja, dit lukt, omdat ze andere rijen updaten !

b.

Laat gebruiker PRAKTIJK eerst zijn UPDATE rollbacken en laat hem daarna onderstaande update uitvoeren:

```
UPDATE theorie.medewerkers  
SET provincie=LOWER(provincie)  
WHERE afd_nr=3;
```

Lukt dit? Waarom?

Nee, de update hangt.

De rijen zijn gelockt door de gebruiker THEORIE!

Laat gebruiker THEORIE zijn update ongedaan maken (ROLLBACK)

Wat gebeurt er?

De locks op die rijen worden vrijgegeven.

Laat gebruiker PRAKTIJK zijn update ongedaan maken (ROLLBACK).

ROLLBACK;

c.

Laat gebruiker THEORIE onderstaande update uitvoeren:

```
UPDATE medewerkers  
SET salaris=salaris*1.05  
WHERE afd_nr=3;
```

Laat gebruiker PRAKTIJK onderstaande update uitvoeren:

```
UPDATE theorie.medewerkers  
SET tussenvoegsel=LOWER(tussenvoegsel)  
WHERE UPPER(provincie)='LI';
```

Lukt dit? Waarom?

Nee, want theorie lockte rijen, die praktijk wil updaten!

Laat gebruiker THEORIE zijn update ongedaan maken (ROLLBACK)

Wat gebeurt er?

De locks worden vrijgegeven, en de update van PRAKTIJK wordt uitgevoerd!

Laat gebruiker PRAKTIJK zijn update ongedaan maken (ROLLBACK).

ROLLBACK;

3.

Laat gebruiker THEORIE de onderstaande instructie geven:

```
SELECT *  
FROM medewerkers  
WHERE afd_nr=3  
FOR UPDATE NOWAIT;
```

Wat gebeurt er?

De rijen worden op voorhand gelockt!

Laat gebruiker PRAKTIJK onderstaande instructie geven:

```
SELECT *  
FROM theorie.medewerkers  
WHERE afd_nr=3  
FOR UPDATE NOWAIT;
```

Wat gebeurt er?

Error: ORA-00054: resource busy and acquire with NOWAIT specified or timeout expired

Laat gebruiker PRAKTIJK onderstaande instructie geven:

```
SELECT *  
FROM theorie.medewerkers  
WHERE salaris<=25000  
FOR UPDATE NOWAIT;
```

Wat gebeurt er?

Gebruiker PRAKTIJK probeert ook dezelfde rijen op voorhand te locken, maar die zijn al gelockt!

Laat gebruiker THEORIE en PRAKTIJK een rollback geven om hun locks terug vrij te geven.

ROLLBACK;

4.

a. Maak een nieuwe afdeling aan met volgende karakteristieken

afd_nr: 10

afd_naam: Kaaiman Eilanden

Selecteer de afdeling ter verificatie.

```
INSERT INTO afdelingen (afd_nr,afd_naam )  
VALUES(10,'Kaaiman Eilanden');  
SELECT * FROM afdelingen  
WHERE afd_nr = 10;
```

b. Stel een savepoint in.

```
SAVEPOINT afdeling_aangemaakt;
```

c. Transfereer iedereen zonder manager én iedereen die rechtstreeks voor de directeur, met medewerker nummer 999666666, werkt naar de nieuwe afdeling, met volgende bewerking:

```
UPDATE medewerkers  
SET afd_nr = 10  
WHERE mgr_sofi_nr in (999666666,null);
```

Selecteer de medewerkers van de nieuwe afdeling, zodat je onderstaand resultaat krijgt:

	SOFI_NR	ACHTERNAAM	MGR_SOFI_NR
1	999555555	Jochems	999666666
2	999444444	Zuiderweg	999666666

```
SELECT sofi_nr, achternaam, mgr_sofi_nr
FROM medewerkers
WHERE afd_nr = 10;
```

- d. Er is een probleempje: de directeur heeft zelf geen baas, maar is niet mee verplaatst. Hoe komt dit?

Een test ten opzichte van een NULL waarde (behalve IS NULL) geeft nooit een positief resultaat. De IN operator heeft dus de NULL waarden niet geselecteerd.

- e. Draai de verplaatsing snel terug, maar zorg ervoor dat de nieuwe afdeling blijft bestaan. Verifieer met de eerder gedane selects dat aan beide voorwaarden voldaan is.

```
ROLLBACK TO afdeling_aangemaakt;
```

- f. Verplaats nu de directeur en zijn medewerkers met de correcte updatebewerking.

```
UPDATE medewerkers
SET afd_nr = 10
WHERE mgr_sofi_nr = 999666666 OR mgr_sofi_nr IS NULL;
```

- g. Maak alle veranderingen, inclusief het aanmaken van de afdeling, ongedaan.

```
ROLLBACK;
```

Dit is het einde van deze oefening reeks. Laat, tot slot, de gebruiker THEORIE de gewijzigde gegevens herstellen:

```
UPDATE medewerkers
SET plaats=InitCap(plaats)
WHERE afd_nr=7;
```

