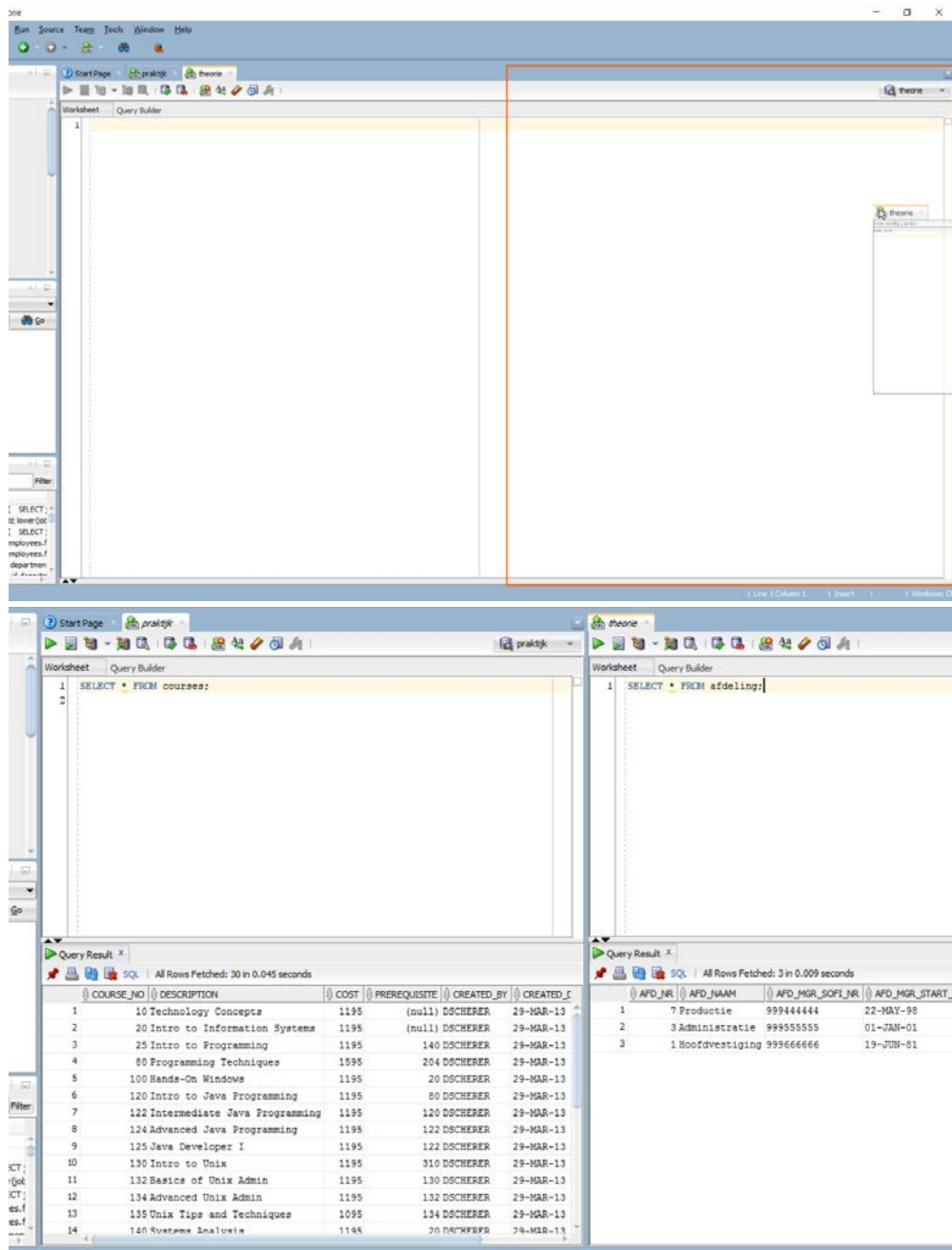


Oefeningen op selects en DML in een multi user omgeving

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?

Hoe schrijf je de instructie correct?

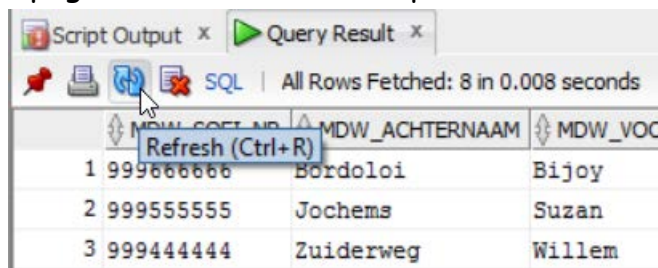
Ziet gebruiker PRAKTIJK de wijziging die door gebruiker THEORIE is doorgevoerd?

Hoe komt dat?

Laat gebruiker THEORIE COMMIT geven

Laat gebruiker PRAKTIJK opnieuw selecteren op de tabel MEDEWERKER.

Tip: gebruik de refresh knop:



	MDW_COEF_NO	MDW_ACHTERNAAM	MDW_VOORNAAM
1	999888888	Bordoloi	Bijoy
2	999555555	Jochems	Suzan
3	999444444	Zuiderweg	Willem

Conclusie:

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?

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?

Laat gebruiker THEORIE zijn update ongedaan maken (ROLLBACK)

Wat gebeurt er?

Laat gebruiker PRAKTIJK zijn update ongedaan maken (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?

Laat gebruiker THEORIE zijn update ongedaan maken (ROLLBACK)

Wat gebeurt er?

Laat gebruiker PRAKTIJK zijn update ongedaan maken (ROLLBACK).

3.

Laat gebruiker THEORIE de onderstaande instructie geven:

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

Wat gebeurt er?

Laat gebruiker PRAKTIJK onderstaande instructie geven:

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

Wat gebeurt er?

Laat gebruiker PRAKTIJK onderstaande instructie geven:

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

Wat gebeurt er?

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

4.a. Maak een nieuwe afdeling aan met volgende karakteristieken

afd_nr: 10
afd_naam: Kaaiman Eilanden

Selecteer de afdeling ter verificatie.

a. Stel een savepoint in.

b. 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

- c. Er is een probleempje: de directeur heeft zelf geen baas, maar is niet mee verplaatst. Hoe komt dit?
- d. 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.
- e. Verplaats nu de directeur en zijn medewerkers met de correcte updatebewerking.
- f. Maak alle veranderingen, inclusief het aanmaken van de afdeling, ongedaan.

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;
```