**Data retrieval language**

* select

**DML, Data Manipulation Language**

* insert
* update
* delete

**DDL, Data Definition Language**

* create
* alter
* drop

**DCL, Data Control Language**

* grant
* revoke

**TCL, Transaction Control Language**

* commit
* rollback

**create user (name)**

**identified by “password”**

**default tablespace data**

**temporary tablespace temp**

**quota 24576 k (lagringsutrymme, 24576 är bits, undgefär 24 mb)**

**on data**

**account unlock;**

**select tablespace\_name, file\_name**

**from dba\_data\_files**

**order by tablespace\_name;**

**alter system set DB\_CREATE\_FILE\_DEST = ‘C:** **\ORACLE12SA\ORADATA\OSRVDB\DATAFILE’;**

**För att skapa ett nytt tablespace i den destinationen (instansen (som är databasen då)).**

**Skapa tablespace:**

**create tablespace data**

**extent management local**

**segment space management auto**

**online;**

**lägg till datafiler I respektive tablespace**

**lägg till datafiler:**

**alter tablespace data**

**add datafile ’data01.dbf’**

**size 1024m;**

**alter tablespace data**

**add datafile ’data02.dbf’**

**size 1024m;**

**när man kör denna select sats igen:**

**select tablespace\_name, file\_name**

**from dba\_data\_files**

**order by tablespace\_name;**

**så kan man kontrollera att tablespace och datafiles skapades**

**PK (primärnyckel), ska aldrig någonsin ändras**

**FK, Foreign key eller främmande nyckel**

* + Kallas även för Referential integrity eller referensintegritet
  + Tabeller kan **relateras till varandra**genom använda en *foreign key constraint*
  + En **FK** placeras på en kolumn i *child-tabellen*
  + Som en del av constraintdefinition i child-tabellen, så refereras en kolumn i en annan tabell, *parent-tabellen*
  + \*Om en rad stoppas in i child-tabellen, med ett **icke-null-värde** i FK-kolumnen, så **måste** det finnas en rad i parent tabellen som har **samma värde** i den refererade kolumnen (se bild, s.18 i Relationsdatabaskomponenter.pdf, 42:18 på respektive video)

lägga in en rad:

**insert into personal (anstnr, fnamn, enamn, lön, avdnr)**

**values (158, ’Arman’, ’Kricic’, 36500, ’400’);**

**unique constraint (ett värde som aldrig kan repeteras, men det kan vara null)**

**Check:**

**check (lön between 15000 and 150000)**

**check constraint (ett värde som inte kan hamna utanför ett visst intervall)**

**not null contstraint (ett värde kan aldrig vara null (odefinierat))**

**create table student(**

**studnr varchar2(10),**

**fnamn varchar2(50) not null,**

**enamn varchar2(50) not null,**

**mobil varchar2(15));**

**alter table student**

**add primary key(studnr);**

**desc user\_constraints;**

**select constraint\_name,constraint\_type**

**from user\_constraints**

**where table\_name = 'STUDENT'; -- OBS! Stora bokstäver**

**Byta namn på constraints**

**alter table student**

**rename constraint SYS\_C0010189 to student\_studnr\_pk;**

**Ta bort constraint**

**alter table student**

**drop constraint student\_studnr\_pk;**

**Nu är PK borttagen. Om vi vill kan vi lägga till fler constraints på**

**samma gång. Vi lägger till PK igen och en unique constraint på**

**kolumnen mobil:**

**alter table student**

**add constraint student\_studnr\_pk primary key(studnr)**

**add constraint student\_mobil\_uq unique(mobil);**

**Lägga till en FK**

**create table ägarbyte(**

**id number(9),**

**persnr varchar2(11),**

**regnr varchar2(6),**

**datum date default sysdate not null);**

**alter table ägarbyte**

**add constraint ägarbyte\_radnr\_pk primary key(id)**

**add constraint ägarbyte\_persnr\_fk foreign key(persnr) references person(persnr)**

**add constraint ägarbyte\_regnr\_fk foreign key(regnr) references fordon(regnr);**

**Lägga till en kolumn**

**alter table student**

**add persnr number(10);**

**Oj, det blev visst fel datatyp på kolumnen. Vi vill ha**

**varchar2(11) istället så att vi kan lagra personnummer**

**som en teckensträng på formen 'YYMMDD‐NNNN'.**

**alter table student**

**modify persnr varchar2(11);**

**create sequence seq\_ägarbyte**

**start with 1**

**increment by 1;**

**kolla att det är rätt**

**select seq\_ägarbyte.nextval**

**from dual;**

**drop object\_name för att ta bort**

**to\_char, to\_date, to\_number etc.**

**Vi kan konvertera ett datum, t. ex. sysdate till en teckensträng på önskat**

**format 2027‐10‐01:14:15:58.**

**Vi testar med en bekant SQL‐sats:**

**select to\_char(sysdate,'YYYY-MM-DD:HH24:MI:SS') as nu**

**from dual;**

**NU**

**-------------------**

**2015-03-12:13:15:49**

**Vi måste se till att det är samma datatyp på bägge sidor om likhetsoperatorn =**

**where to\_char(regdatum,'YYYY-MM-DD') = '2019-05-25';**

**Om vi vill kontrollera hur många som registrerade sig under 2019:**

**where to\_char(regdatum,'YYYY’) = '2019';**

**Om vi vill kontrollera hur många som registrerade sig under Augusti 2019:**

**where to\_char(regdatum,'YYYY-MM') = '2019-08';**

**antalet dagar mellan två datum**

**select to\_date('JAN-01-2026','MON-DD-YYYY') + 10 as återdatum**

**from dual;**

**ÅTERDATUM**

**----------------**

**2026-01-11**

**select sysdate - to\_date('1993-02-23','YYYY-MM-DD')**

**from dual;**

**resultat:**

**sysdate - to\_date('1993-02-23','YYYY-MM-DD')**

**--------------------------------------------**

**antalet dagar: 8052,74644**

**Avrunda decimaler**

**select round(sysdate - to\_date('1993-02-23','YYYY-MM-DD'),0) as antal\_dagar**

**from dual;**

**ANTAL\_DAGAR**

**------------**

**8053**

**Vi kan använda funktionen round() för att avrunda till önskvärt antal**

**decimaler:**

**Här kan du läsa mer om Oracle built in functions:**

[**http://www.techonthenet.com/oracle/functions/**](http://www.techonthenet.com/oracle/functions/)

**Anchored declaration**

**declare**

**v\_email kund.email%type;**

**v\_mobil kund.mobil%type;**

**^ ^**

**tabell kolumn**

**Hämtar datatypen från databasen.**

**declare**

**v\_email varchar2(30);**

**v\_mobil varchar2(10);**

**begin**

**select email, mobil**

**into v\_email, v\_mobil**

**from kund**

**where knr = 56482;**

**dbms\_output.put\_line(‘mobil: ’ || v\_mobil || ‘,’ || ‘email: ‘ || v\_email);**

**end;**

**/**

**select length(‘something’)**

**from dual;**

**Deklarera en cursor**

**declare**

**cursor c\_kunder is select fnamn,enamn,adress**

**from kund;**

**begin**

**null;**

**end;**

**declare**

**cursor c\_storkunder is**

**select fnamn,enamn,adress**

**from kund**

**where knr in(select knr**

**from kundorder**

**where ordnr in(select orderrad.ordnr**

**from orderrad,artikel**

**where orderrad.artnr = artikel.artnr**

**having sum(artikel.pris \* orderrad.antal) > 20000**

**group by orderrad.ordnr));**

**Cursor loop**