Databashanterare

Programvara för att

- skapa,
- utforma (designa),
- lagra och
- administrera databaser,

Kallas *Database management system (DBMS eller RDBMS)* som i regel installeras på en server.

- Exempel på *databashanterare* är *Access, (Excel), SQL-Server, Oracle 21c, MySQL, DB2, FoxPro, Paradox, …*

- Alla DBMS har stöd för *SQL*, men även för procedurala utökningar av SQL:

  Oracle *PL/SQL*,

  Micorsoft *Transact SQL, …*

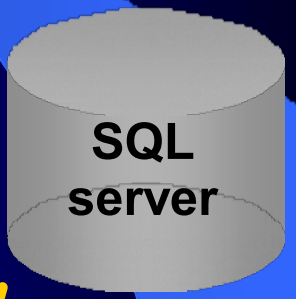
Klient - Server modellen

SQL ställer en fråga, t.ex.:

```
SELECT department_name
FROM   departments;
```

Frågan skickas till servern

dvs ber om en tjänst *(klient)*, beskriver vad den vill ha, men inte hur svaret kommer till.



Servern svarar

DEPARTMENT_NAME
Administration
Marketing
Shipping
IT
Sales
Executive
Accounting
Contracting

Operationens slutresultat är alltid en tabell: *”Resultattabell”*

SQL – databasers språk

Structured Query Language

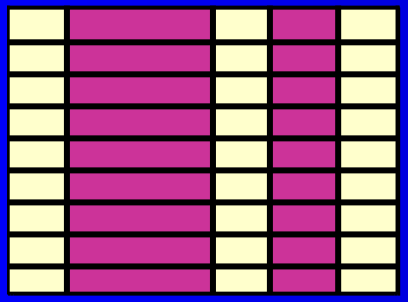
(Strukturerat frågespråk)

- Standardspråk för kommunikation med relationsdatabaser.
- Oberoende av databashanterare.
- Utvecklades på 70-talet av IBM.
  Idag: allmän standard, senaste version: SQL-99
- Med SQL kan man ställa *”frågor”* till databaser för att
  - ta fram, uppdatera,
  - sortera och
  - strukturera information i databaser,
  - skapa tabeller, definiera constraints
  - ge rättigheter till databasobjekt, …

SELECT-*satsen*

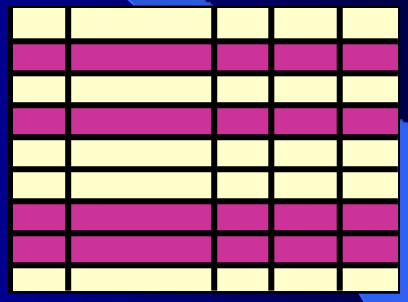
Läser från databasen och visar data i kolumner och rader:

1. Projektion



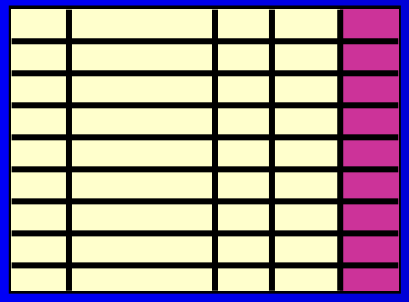
Selekterar kolumner från 1 tabell

2. Selektion

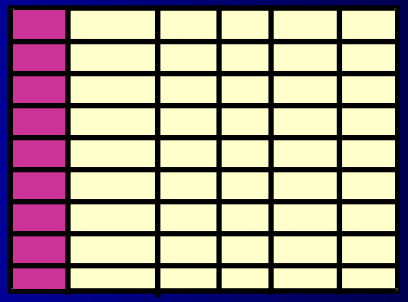


Selekterar rader från 1 tabell

Tabell 1



3. Join
(Kap. 3)



Tabell 2

Tar ut data från flera tabeller

Att ta ut alla kolumner

```
SELECT *
FROM   departments;
```

DEPARTMENT_ID	DEPARTMENT_NAME	MANAGER_ID	LOCATION_ID
10	Administration	200	1700
20	Marketing	201	1800
50	Shipping	124	1500
60	IT	103	1400
80	Sales	149	2500
90	Executive	100	1700
110	Accounting	205	1700
190	Contracting		1700

...

SELECT talar om vilka kolumner som ska tas ut.
FROM talar om från vilken tabell kolumnerna ska tas ut.

Ger samma resultat:

```
SELECT department_id, department_name, manager_id, location_id
FROM   departments;
```

Att ta ut vissa kolumner

```
SELECT department_id, location_id
FROM   departments;
```

DEPARTMENT_ID	LOCATION_ID
10	1700
20	1800
50	1500
60	1400
80	2500
90	1700
110	1700
190	1700

...

Kommaseparerad lista över kolumner som anger ordningen endast i den aktuella *utskriften*, inte i databasen.

SELECT-satsen är read-only, kan inte ändra databasen.

Hittills har vi endast använt projektion: Att selektera kolumner.

Att selektera rader: selektion

Tabellen `EMPLOYEES`

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
104	Ernst	IT_PROG	60
107	Lorentz	IT_PROG	60
124	Mourgos	ST_MAN	50

...

Vilka anställda jobbar på avd. 90?

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

För att få ut det måste ett villkor läggas till **`SELECT`**-satsen.

Detta görs med den nya satsdelen (eng. *clause*) **`WHERE`**.

Att lägga till villkor med `WHERE`

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees
WHERE  department_id = 90 ;
```

EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90

Enkelt *villkor* på likhet mellan tal.
= är här en jämförelseoperator.

OBS! **Villkoret kan involvera en kolumn som inte ens förekommer i `SELECT`-satsen:**

```
SELECT employee_id, last_name, job_id
FROM   employees
WHERE  department_id = 90 ;
```

Radsortering med ORDER BY

```
SELECT   last_name, job_id, department_id, hire_date
FROM     employees
ORDER BY hire_date ;
```

LAST_NAME	JOB_ID	DEPARTMENT_ID	HIRE_DATE
King	AD_PRES	90	17-JUN-87
Whalen	AD_ASST	10	17-SEP-87
Kochhar	AD_VP	90	21-SEP-89
Hunold	IT_PROG	60	03-JAN-90
Ernst	IT_PROG	60	21-MAY-91

...

Satsdelen **ORDER BY** kommer sist i **SELECT**-satsen.
Utan **ORDER BY** är radordningen odefinierad.

ORDER BY sorterar by default i stigande ordning dvs ASC (ASCending).
För fallande ordning kan DESC (DESCending) användas:

```
SELECT   last_name, job_id, department_id, hire_date
FROM     employees
ORDER BY hire_date DESC;
```

CREATE TABLE-*satsen*

```
CREATE TABLE Kurser
(
   KursID    INT          IDENTITY (1,1) NOT NULL,
   Namn      VARCHAR(50)                 NOT NULL,
   Längd     INT                             NULL,
   InstID    INT                         NOT NULL
);
```

- Måste specificeras:
  - Tabellnamn: **`Kurser`**
  - Kolumnnamn: **`KursID, Namn, Längd, InstID`**
  - Kolumndatatyper: **`INT, VARCHAR(50)`**

  (OBS! Inte optional)

Regler & konventioner

Några regler för SQL-satser:

- SQL-satser är inte case sensitive.
- Det är inte obligatoriskt att avsluta SQL-satser med semikolon.
- SQL-satser kan skrivas på en eller flera rader.
- Reserverade ord kan ej förkortas.

Konventioner:

- Skriv reserverade ord med versaler.
- Börja reserverade ord på separat rad.
- Avsluta *SQL*-satserna med semikolon.