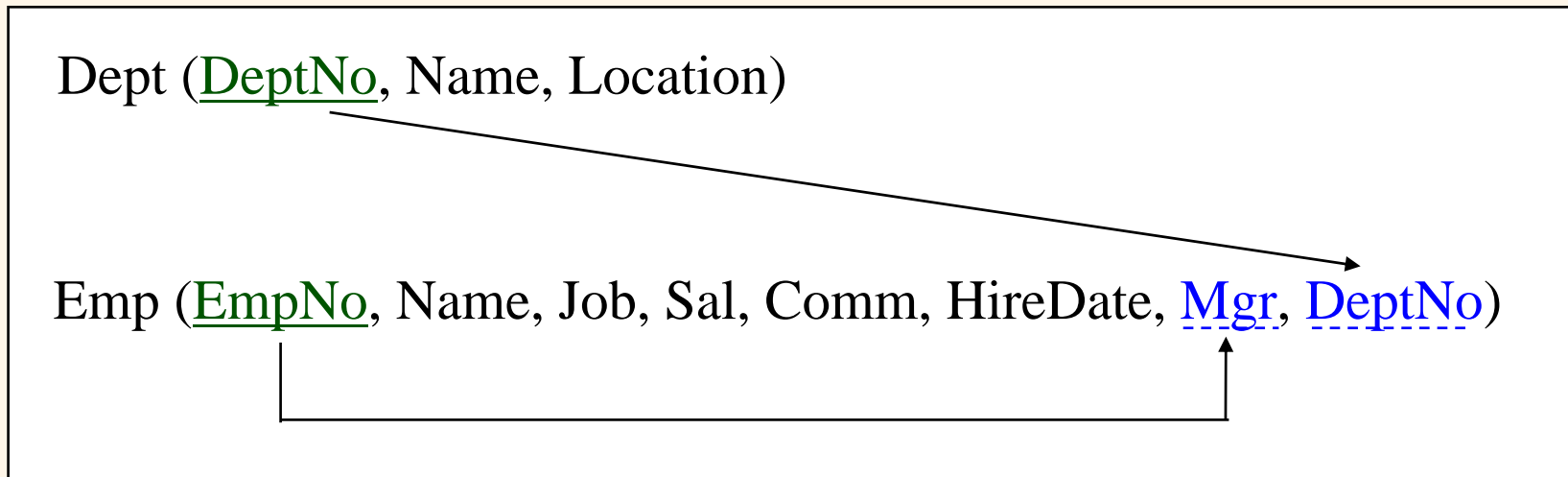
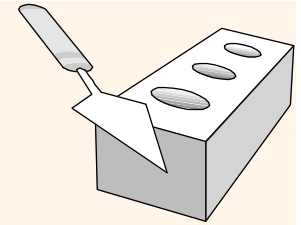


SQL: Update

Example Schema



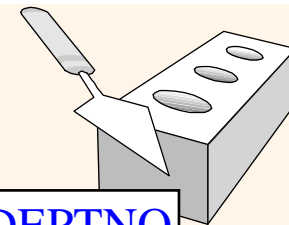
Dept

<u>DeptNo</u>	Number(2)
Name	Varchar2(14)
Location	Varchar2(13)

Emp

<u>EmpNo</u>	Number(4)
Name	Varchar2(10)
Job	Varchar2(9)
Sal	Number(7,2)
Comm	Number(7,2)
HireDate	Date
<u>Mgr</u>	Number(4)
<u>DeptNo</u>	Number(2)

Example Relations



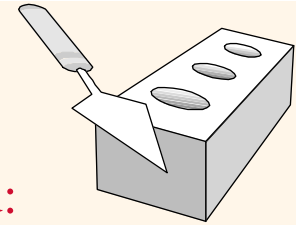
Emp

EMPNO	NAME	JOB	MGR	HIREDATE	SAL	COMM	DEPTNO
7369	SMITH	CLERK	7902	17-DEC-80	800		20
7499	ALLEN	SALESMAN	7698	20-FEB-81	1600	300	30
7521	WARD	SALESMAN	7698	22-FEB-81	1250	500	30
7566	JONES	MANAGER	7839	02-APR-81	2975		20
7654	MARTIN	SALESMAN	7698	28-SEP-81	1250	1400	30
7698	BLAKE	MANAGER	7839	01-MAY-81	2850		30
7782	CLARK	MANAGER	7839	09-JUN-81	2450		10
7788	SCOTT	ANALYST	7566	19-APR-87	3000		20
7839	KING	PRESIDENT		17-NOV-81	5000		10
7844	TURNER	SALESMAN	7698	08-SEP-81	1500	0	30
7876	ADAMS	CLERK	7788	23-MAY-87	1100		20
7900	JAMES	CLERK	7698	03-DEC-81	950		30
7902	FORD	ANALYST	7566	03-DEC-81	3000		20
7934	MILLER	CLERK	7782	23-JAN-82	1300		10

Dept

DEPTNO	NAME	LOCATION
10	ACCOUNTING	NEW YORK
20	RESEARCH	DALLAS
30	SALES	CHICAGO
40	OPERATIONS	BOSTON

INSERT Statement



To create a tuple in SQL the following 'Insert' command is required:

```
insert into R (attribute1, attribute2, ... attributen)  
values (value1, value2, ... valuen)
```

```
insert into Dept (deptno, name, location )  
values (10, 'Accounting', 'New York')
```

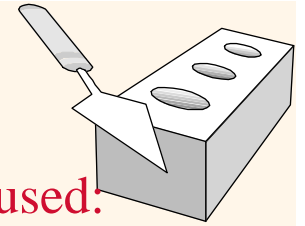
```
insert into Dept (deptno, name, location )  
values (30, 'Sales', 'Chicago')
```

*The insert order matters in
terms of referential integrity
constraints!*

```
insert into Emp (empno, name, job, sal, comm, hiredate, mgr, deptno )  
values (7839, 'King', 'President', 5000, NULL, '17-Nov-81', NULL, 10)
```

```
insert into Emp (empno, name, job, sal, comm, hiredate, mgr, deptno )  
values (7698, 'Blake', 'Manager', 1600, NULL, '01-May-81', 7839, 30)
```

Group INSERT Statement



To create a set of tuples in SQL the following 'Insert' command can be used!

```
insert into R (attribute1, attribute2, ... attributen)
  select (attribute1, attribute2, ... attributen)
  from relation1, relation2, ... relationn
  [where condition-expression]
  [group by attribute1, attribute2, ... attributen]
  [having condition-expression]
  [order by attribute1, attribute2, ... attributen]
```

Example: copy details of all employees that work in department 10 from the Emp relation into the DepA relation.

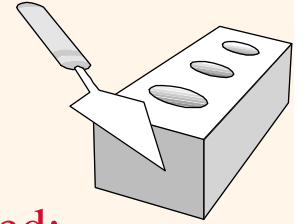
```
insert into DepA (staffno, name, job, hiredate)
  select empno, name, job, hiredate
  from Emp
  where deptno = 10;
```



corresponding
attributes have to
be of the same type

Each tuple to be inserted has to be unique!

DELETE Statement



To delete a set of tuples in SQL the following 'Delete' command is used:

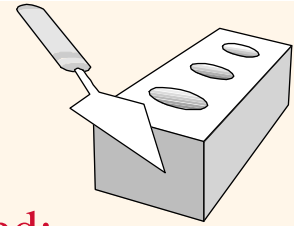
```
delete from R  
[where condition-expression]
```

Example: remove details of all employees that work in department 10 from the Emp relation.

```
Delete from Emp  
where deptno = 10;
```

*If the where clause is omitted then **all** tuples in the relation will be removed!*

UPDATE Statement



To alter a set of tuples in SQL the following 'Update' command is used:

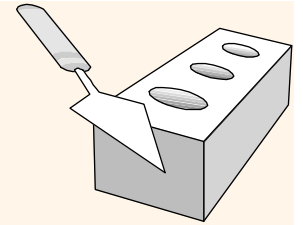
```
update R
set attribute1 = datavalue1,
    attribute2 = datavalue2, ...
    attributen = datavaluen
[where condition-expression]
```

Example: increase the salary of all employees that work in department 10 by 10%.

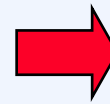
```
update Emp
set sal = sal * 1.1
where deptno = 10;
```

*If the where clause is omitted then **all** tuples in the relation will be altered!*

ORDER BY

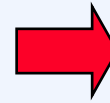


```
select emp.name, dept.name
from Emp, Dept
where (emp.deptno = dept.deptno)
      and (emp.deptno = 10 or emp.deptno = 30)
order by emp.name asc;
```



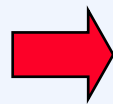
Blake	Sales
King	Accounting

```
select emp.name, dept.name
from Emp, Dept
where (emp.deptno = dept.deptno)
      and (emp.deptno = 10 or emp.deptno = 30)
order by dept.name desc;
```



Blake	Sales
King	Accounting

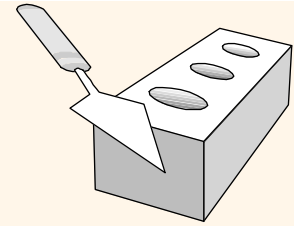
```
select name
from Dept
order by name;
```



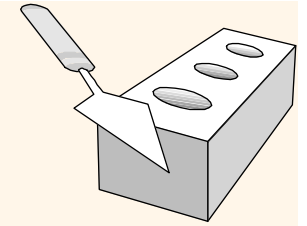
Accounting
Sales

*Remember in relations
neither tuples nor attributes
have any intrinsic order!*

Views in SQL



- ❖ A SQL view is a virtual table that is **derived** from other base or virtual tables
- ❖ Base tables are defined by CREATE TABLE command and are permanently stored in a database
- ❖ Virtual tables are defined by the CREATE VIEW command to avoid defining complex SQL retrieval expressions repeatedly
- ❖ The definition of a view is stored in the Catalog, but it is not stored in the database itself, so it is **computed** every time it is used in a query



Example

- ❖ A possible view definition

```
CREATE VIEW StudOccupied AS  
  SELECT g.StudId, SUM(Hours) AS Occupied  
  FROM Grades g, Course p  
  WHERE g.CourId = p.CourId AND Grade IS NULL  
  GROUP BY StudId ;
```

- ❖ Deleting a view

```
DROP VIEW StudOccupied ;
```