Referential Integrity Constraints

Integrity Constraints in SQL

1) NOT NULL CONSTRAINT

to disallow the null values

2) Table Constraints

Primary key

Check

Not null

Foreign key

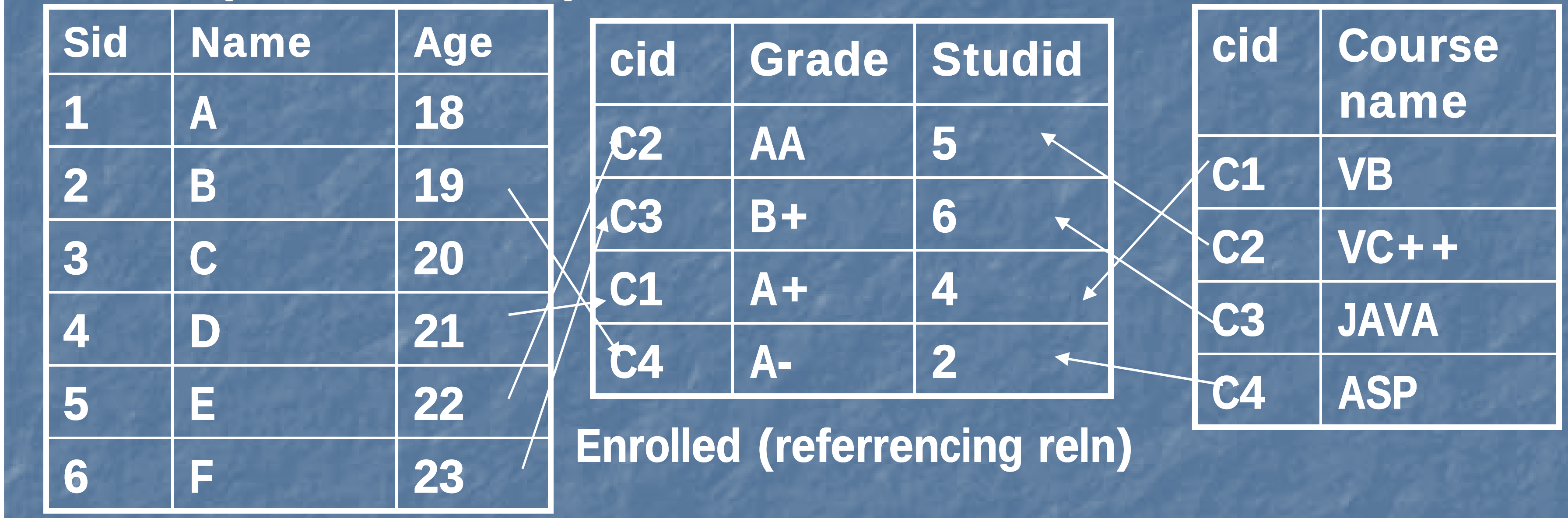
Example

```
Create table Sailors (sid number,
        sname char(10) not null,
        rating number, age number(5,2),
        primary key (sid),
        check (rating >=1 and rating <=10)
Create table Reserves (sid number, bid number,
                          day date,
      foreign key (sid) references sailors (sid),
      foreign key (bid) references boats (bid),
      constraint noInterLake
      check (‘InterLake’ <>(select B.bname from
      boats B where b.bid = reserves.bid))
```

3) Domain Constraints and distinct types

Create domain ratingval number default 1 check (value >=1 and value <=10)

Referrential Integrity Constraint



Students (referrenced reln)

Sid	Name	Age
1	A	18
2	B	19
3	C	20
4	D	21
5	E	22
6	F	23

Enrolled (referrencing reln)

cid	Grade	Studid
C2	AA	5
C3	B+	6
C1	A+	4
C4	A-	2

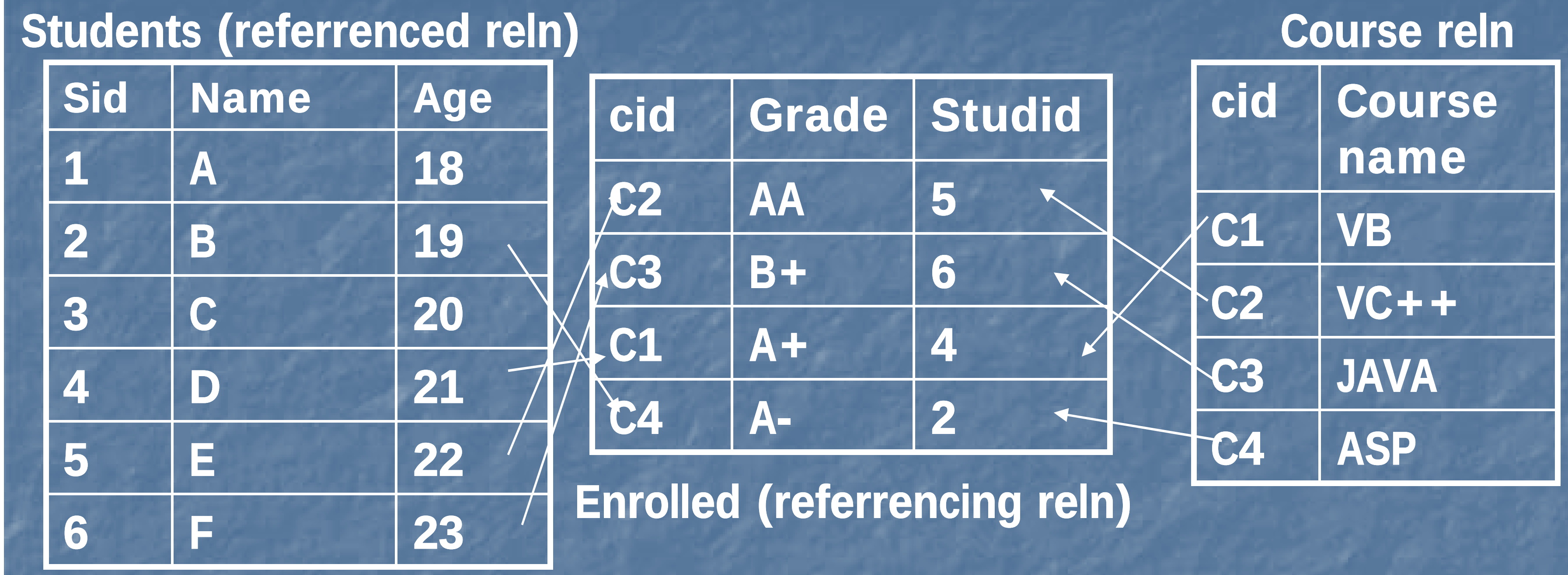
Course reln

cid	Course name
C1	VB
C2	VC++
C3	JAVA
C4	ASP

Studid and sid must be domain compatible (should have same datatype)

Similarly data type of cid of course reln should be same as datatype of cid of enrolled reln

Referrential Integrity Constarint



Students (referrenced reln)

Sid	Name	Age
1	A	18
2	B	19
3	C	20
4	D	21
5	E	22
6	F	23

Enrolled (referrencing reln)

cid	Grade	Studid
C2	AA	5
C3	B+	6
C1	A+	4
C4	A-	2

Course reln

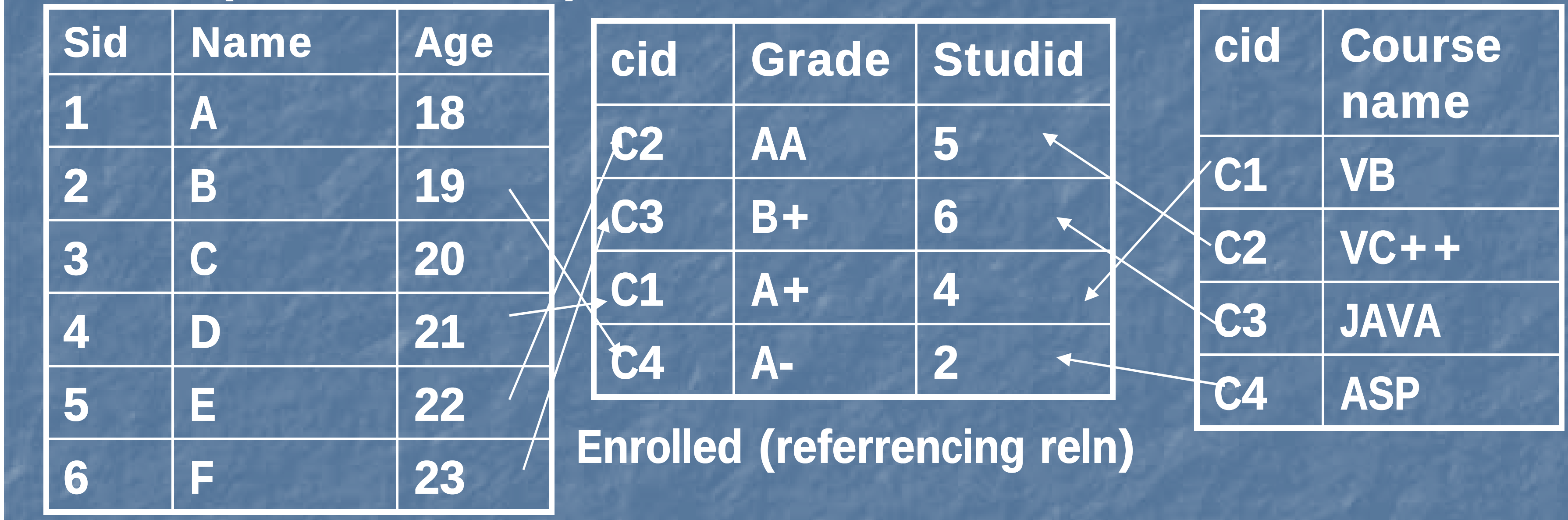
cid	Course name
C1	VB
C2	VC++
C3	JAVA
C4	ASP

Insert (c1, A+, 7) in Enrolled reln is invalid becoz studid <> sid

Foreign key constraint will check for this violation

Delete (6, F, 23) from student reln should also delete (c3, B+, 6) from enrolled reln

Referrential Integrity Constarint



Students (referrenced reln)

Sid	Name	Age
1	A	18
2	B	19
3	C	20
4	D	21
5	E	22
6	F	23

cid	Grade	Studid
C2	AA	5
C3	B+	6
C1	A+	4
C4	A-	2

Enrolled (referrencing reln)

Course reln

cid	Course name
C1	VB
C2	VC++
C3	JAVA
C4	ASP

Null values are not allowed in primary key

Null values are allowed in foreign key

Every studid value in enrolled must appear in student reln since enrolled references student .

```
Create table Enrolled ( studid char(10),
                        cid char (10),
                        grade char (10),
foreign key (studid) references students(sid)
foreign key (cid) references course (cid)
on delete cascade,
on update no action);
```

Options are specified as part of foreign key declaration.

Default option – No action ie reject the action (del or update)

Similarly on update cascade

On delete set default. Eg sid char(20) default ’50’

On delete set null

Cascading Actions in SQL

```
create table account
    . . .
    foreign key(branch-name)   references branch
                               on delete cascade
                               on update cascade
    . . . )
```

- Due to the on delete cascade clauses, if a delete of a tuple in *branch* results in referential-integrity constraint violation, the delete "cascades" to the *account* relation, deleting the tuple that refers to the branch that was deleted.
- Cascading updates are similar.

Cascading Actions in SQL (Cont.)

- If there is a chain of foreign-key dependencies across multiple relations, with on delete cascade specified for each dependency, a deletion or update at one end of the chain can propagate across the entire chain.
- If a cascading update to delete causes a constraint violation that cannot be handled by a further cascading operation, the system aborts the transaction.
  - As a result, all the changes caused by the transaction and its cascading actions are undone.
- Referential integrity is only checked at the end of a transaction
  - Intermediate steps are allowed to violate referential integrity provided later steps remove the violation
  - Otherwise it would be impossible to create some database states, e.g. insert two tuples whose foreign keys point to each other
    - E.g. *spouse name* attribute of relation
      marriedperson(name, address, spousename)

- Alternative to cascading:
  - on delete set null
  - on delete set default
- Null values in foreign key attributes complicate SQL referential integrity semantics, and are best prevented using not null
  - if any attribute of a foreign key is null, the tuple is defined to satisfy the foreign key constraint!

Deferred constraint

Constraint :

Every student is required to have an honors course and Every course is required to have a grader who is some student.

Create table student (sid char(20), name char (20), age number, honors char(10) not null, primary key (sid), foreign key (honors) references course (cid));

Create table course (cid char(10), cname char(10), credits number, grader char(20) not null, primary key (cid), foreign key (grader) references students (sid))

Problem in inserting first tuple in both tables. One cannot insert without other.

So defer the insertion

Sql allows the constraint to be in deferred or immediate mode

Set constraint XYZ deferred. This constraint will be checked during commit.

Group by and Having clause

Find the age of the youngest sailor who is eligible to vote for each rating level with at least two such sailors.

Sid	Sname	Rating	Age
22	A	7	45
24	B	1	33
31	C	8	55
32	D	8	22
33	E	7	35
34	F	3	25
35	G	3	63
36	H	3	21
37	I	3	10

Rating	Age
1 X	33
3	25
3	63
3	21
7	45
7	35
8	55
8	22

Group by and Having clause

Find the age of the youngest sailor who is eligible to vote for each rating level with at least two such sailors.

```
Select s.rating, MIN (s.age) as minage
From sailor s
Where s.age>=18
Group by s.rating
Having count(*) > 1
```