CONSTRAINTS

- There are three *main types* of constraints in the relational model:
  1. **Key** constraints
  2. **Entity integrity** constraints
  3. **Referential integrity** constraints

14

3. REFERENTIAL INTEGRITY Constraints

A constraint involving **two** relations

Used to specify a **relationship** among tuples in two relations. The **referencing relation** and the **referenced relation**.

15

3. REFERENTIAL INTEGRITY Constraints

The **value** in the **foreign key** column (or columns) **FK** of the the **referencing relation** R1 can be **either** a **value** of an **existing primary key** value of a corresponding primary key PK in the **referenced relation** R2, **OR a NULL**
If Null, the **FK** in **R1** **should not** be a part of **its own primary key.**

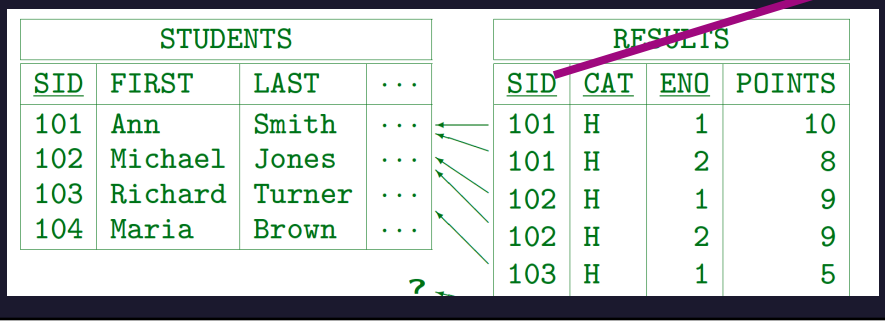
16

Foreign key

- **A FOREIGN KEY** is a key used to link two tables together.
- **A FOREIGN KEY** is a field (or collection of fields) in one table that refers to the **PRIMARY KEY** in another table.
- The table containing the foreign key is called the **referencing** table, and the table containing the candidate key is called the **referenced** table.
- A tuple t1 in R1 is said to **reference** a tuple t2 in R2 if t1[FK] = t2[PK].

Foreign Key Reference SID

SID in **RESULTS** is a **foreign key** referencing **STUDENTS**:



STUDENTS

SID	FIRST	LAST	...
101	Ann	Smith	...
102	Michael	Jones	...
103	Richard	Turner	...
104	Maria	Brown	...

RESULTS

SID	CAT	ENO	POINTS
101	H	1	10
101	H	2	8
102	H	1	9
102	H	2	9
103	H	1	5

17

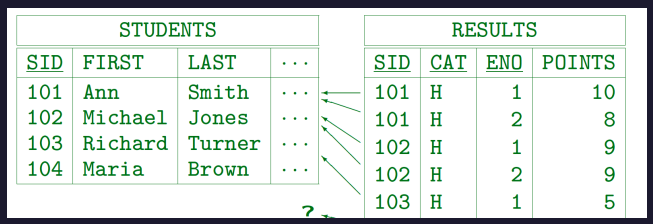
3. REFERENTIAL INTEGRITY Constraints

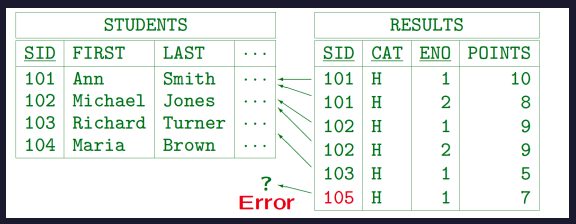
The **value** in the **foreign key** column (or columns) **FK** of the the **referencing relation** R1 can be **either** a **value** of an **existing primary key** value of a corresponding primary key PK in the **referenced relation** R2, **OR a NULL**
If Null, the **FK** in **R1** **should not** be a part of **its own primary key.**

18

REFERENTIAL INTEGRITY Constraints

- The foreign key constraint ensures that for every tuple t in RESULTS there is a tuple u in STUDENTS such that t.SID = u.SID
- The constraint that is needed here is that every SID value in RESULTS also appears in STUDENTS

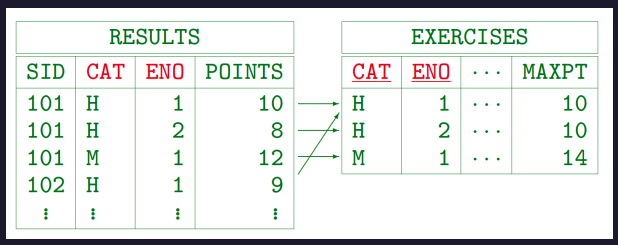




Why?

19

Composite Keys



RESULTS

SID	CAT	ENO	POINTS
101	H	1	10
101	H	2	8
101	M	1	12
102	H	1	9
⋮	⋮	⋮	⋮

EXERCISES

CAT	ENO	···	MAXPT
H	1	···	10
H	2	···	10
M	1	···	14

- A table with a composed key (like EXERCISES) must be referenced with a composed foreign key that has the same number of columns.
- Corresponding columns must have the same data type.
- It is not required that corresponding columns have the same name.
- In the example, the composed foreign key requires that every combination of CAT and ENO which appears in RESULTS, must also appear in EXERCISES

20

3. REFERENTIAL INTEGRITY Constraints

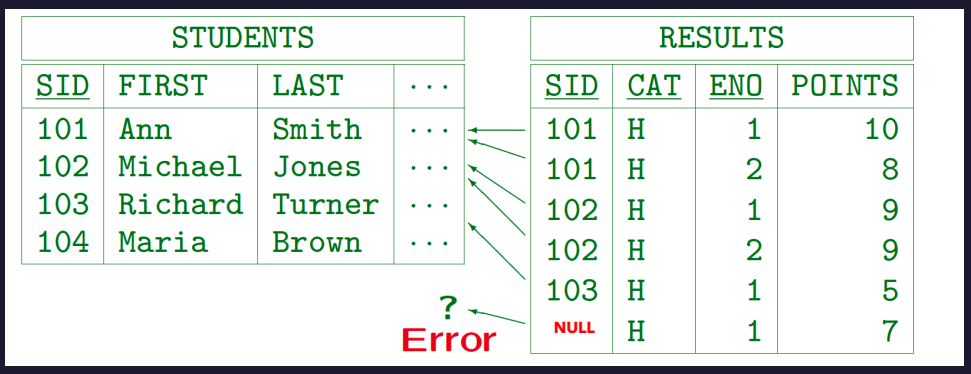
The **value** in the **foreign key** column (or columns) **FK** of the the **referencing relation** R1 can be **either** a **value** of an **existing primary key** value of a corresponding primary key PK in the **referenced relation** R2,

OR a NULL

If Null, the **FK** in **R1** **should not** be a part of **its own primary key.**

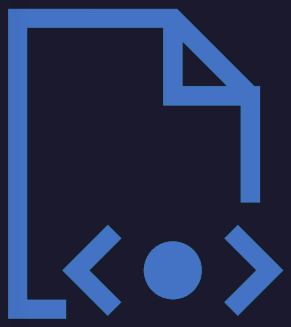
21

If Null, the **FK** in **R1** **should not** be a part of **its own primary key.**



22

Other Constraints



- implicit constraint: the **domain** constraint: Every value in a tuple must be from the *domain of its attribute* (or it could be **null**, if allowed for that attribute)
- **Semantic Integrity** Constraints: **based on application** semantics and **cannot be expressed by the model per se**

23