Student($sid$, name, addr, age, GPA)

<table>
<thead>
<tr>
<th>sid</th>
<th>name</th>
<th>addr</th>
<th>age</th>
<th>GPA</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>John</td>
<td>Ki#Bu!GK.$@q</td>
<td>19</td>
<td>2.1</td>
</tr>
<tr>
<td>301</td>
<td>Elaine</td>
<td>301 Wilshire</td>
<td>263</td>
<td>3.9</td>
</tr>
<tr>
<td>401</td>
<td>James</td>
<td>183 Westwood</td>
<td>17</td>
<td>-1.0</td>
</tr>
<tr>
<td>208</td>
<td>Esther</td>
<td>421 Wilshire</td>
<td>20</td>
<td>3.1</td>
</tr>
</tbody>
</table>

Class($dept$, $cnum$, sec, unit, title, instructor)

<table>
<thead>
<tr>
<th>dept</th>
<th>$cnum$</th>
<th>sec</th>
<th>unit</th>
<th>title</th>
<th>instructor</th>
</tr>
</thead>
<tbody>
<tr>
<td>CS</td>
<td>112</td>
<td>01</td>
<td>03</td>
<td>Modeling</td>
<td>Dick Muntz</td>
</tr>
<tr>
<td>CS</td>
<td>143</td>
<td>01</td>
<td>04</td>
<td>DB Systems</td>
<td>Carlo Zaniolo</td>
</tr>
<tr>
<td>EE</td>
<td>143</td>
<td>01</td>
<td>134</td>
<td>Signal</td>
<td>Dick Muntz</td>
</tr>
<tr>
<td>ME</td>
<td>183</td>
<td>02</td>
<td>05</td>
<td>Mechanics</td>
<td>Susan Tracey</td>
</tr>
</tbody>
</table>

Enroll($sid$, dept, $cnum$, sec)

<table>
<thead>
<tr>
<th>$sid$</th>
<th>dept</th>
<th>$cnum$</th>
<th>sec</th>
</tr>
</thead>
<tbody>
<tr>
<td>301</td>
<td>CS</td>
<td>112</td>
<td>01</td>
</tr>
<tr>
<td>999</td>
<td>CS</td>
<td>143</td>
<td>01</td>
</tr>
<tr>
<td>401</td>
<td>AT</td>
<td>000</td>
<td>00</td>
</tr>
<tr>
<td>303</td>
<td>EE</td>
<td>143</td>
<td>01</td>
</tr>
<tr>
<td>303</td>
<td>CS</td>
<td>112</td>
<td>01</td>
</tr>
</tbody>
</table>
CREATE TABLE Course ( 
    dept CHAR(2) NOT NULL,
    cnum INT NOT NULL,
    sec INT NOT NULL,
    unit INT,
    instructor VARCHAR(30),
    title VARCHAR(30),
    PRIMARY KEY(dept, cnum, sec),
    UNIQUE(dept, cnum, instructor),
    UNIQUE(dept, sec, title) )
Referential Integrity Constraint

CREATE TABLE Enroll (  
  sid INT REFERENCES Student(sid),  
  dept CHAR(2),  
  cnum INT,  
  sec INT,  
  FOREIGN KEY (dept, cnum, sec)  
    REFERENCES Class(dept, cnum, sec)  
    ON DELETE CASCADE  
    ON UPDATE SET NULL )

• Notes
  – Referencing attributes called FOREIGN KEY
  – Referenced attributes must be PRIMARY KEY or UNIQUE
CHECK Constraints

• CHECK(<condition>) in CREATE TABLE
• Example: all CS classes are of 3 units or more.
  – CREATE TABLE Class (  
    dept CHAR(2),  
    cnum INT,  
    unit INT,  
    title VARCHAR(50),  
    instructor VARCHAR(30),  
    CHECK (dept <> 'CS' OR unit >= 3) )  
  – (dept='CS' → unit>3) ≡ (¬(dept='CS') V unit>3)  
• Constraint is checked whenever a tuple is inserted/updated.  
• In SQL92, conditions can be more complex, e.g., with subqueries  
  – In practice, complex CHECK conditions are not supported
General Assertions

• CREATE ASSERTION <assertion name>
  CHECK (<condition>)
    – Constraint on the entire relation or database

• Example: Average GPA >3.0
  CREATE ASSERTION HighGPA
  CHECK (3.0 < (SELECT AVG(GPA) FROM Student))
Trigger: Event-Condition-Action rule

CREATE TRIGGER <name>
<event>
  <referencing clause>  // optional
WHEN (<condition>)  // optional
<action>

- <event>
  - BEFORE | AFTER INSERT ON R
  - BEFORE | AFTER DELETE ON R
  - BEFORE | AFTER UPDATE [OF A1, A2, ..., An] ON R

- <action>
  - Any SQL statement
  - Multiple statements should be enclosed with BEGIN ... END and be separated by ;

- <referencing clause>
  - REFERENCING OLD | NEW TABLE | ROW AS <var>, ...
    - FOR EACH ROW: row-level trigger
    - FOR EACH STATEMENT (default): statement-level trigger