**Query**

**Candidates**

**Input-Output**

**Examples**

**SQL Query**

**Completion**

**Candidate**

**Ranking**

**Query**

**Skeleton**

**Query Skeleton Creation**

**Provide more input-output examples**

**Ranked**

**Queries**

**Select**

**Desirable**

**Query**

|  |  |
| --- | --- |
| **select** | Student.Student\_name, **<Aggregation>** |
| **from** | student, enrolled |
| **where** | student.Student\_key = enrolled.Student\_key  and **<Conditions>** |
| **group by** | Student.Student\_name |
| **having** | **<Conditions>** |

|  |  |
| --- | --- |
| name | score |
| Bob | 4 |
| Dan | 5 |
| Jim | 2 |

Fgg

ff

|  |
| --- |
| name |
| Bob |
| Dan |

**(b)** The output table

**(a)** The input table: student

1. **select** name **from** student **where** score > 3

2. **select name from** student **where** name = ‘Bob’

or name = ‘Dan’

**Select**

**Desirable**

**Queries**

**Query**

**Skeleton**

**Query**

**Candidates**

**Ranked**

**Queries**

**Query**

**Completion**

**Candidate**

**Ranking**

**Input-Output**

**Examples**

**Skeleton Creation**

**Provide more input-output examples**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  | Aggregation features by student\_id | |
| student\_id | course\_id | score | name | level | count(course\_id) | max(course\_id) |
| 1 | 1 | 4 | Adam | senior | 2 | 2 |
| 1 | 2 | 2 | Adam | senior | 2 | 2 |
| 2 | 1 | 3 | Bob | junior | 3 | 3 |
| 2 | 2 | 2 | Bob | junior | 3 | 3 |
| 2 | 3 | 3 | Bob | junior | 3 | 3 |
| 3 | 2 | 1 | Erin | senior | 1 | 2 |
| 4 | 1 | 4 | Rob | junior | 2 | 3  **count(course\_id) >= 3** |
| 4 | 3 | 4 | Rob | junior | 2 | 3 |
| 5 | 2 | 5 | Dan | senior | 3 | 4 |
| 5 | 3 | 2 | Dan | senior | 3 | 4 |
| 5 | 4 | 1 | Dan | senior | 3 | 4 |
| 6 | 2 | 4 | Peter | senior | 2 | 4 |
| 6 | 4 | 5 | Peter | senior | 2 | 4 |
| 7 | 1 | 2 | Sai | senior | 3 | 4 |
| 7 | 3 | 3 | Sai | senior | 3 | 4 |
| 7 | 4 | 4 | Sai | senior | 3 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| student\_id | course\_id | score | name | level |
| 2 | 1 | 3 | Bob | junior |
| 2 | 2 | 2 | Bob | junior |
| 2 | 3 | 3 | Bob | junior |
| 5 | 2 | 5 | Dan | senior |
| 5 | 3 | 2 | Dan | senior |
| 5 | 4 | 1 | Dan | senior |
| 7 | 1 | 2 | Sai | senior |
| 7 | 3 | 3 | Sai | senior |
| 7 | 4 | 4 | Sai | senior |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| student\_id | course\_id | score | name | level |
| 5 | 2 | 5 | Dan | senior |
| 5 | 3 | 2 | Dan | senior |
| 5 | 4 | 1 | Dan | senior |
| 7 | 1 | 2 | Sai | senior |
| 7 | 3 | 3 | Sai | senior |
| 7 | 4 | 4 | Sai | senior |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| student\_id | course\_id(s) | score(s) | name(s) | level(s) |
| 5 | 2, 3, 4 | 5, 2, 1 | Dan | senior |
| 7 | 1, 3, 4 | 2, 3, 4 | Sai | senior |

|  |  |
| --- | --- |
| name | max\_score |
| Dan | 5 |
| Sai | 4  **projection** |

**(e)**

**(d)**

**(c)**

**(b)**

**(a)**

**group by student\_id**

**level = “senior”**

|  |  |  |  |
| --- | --- | --- | --- |
| Column1 | Column2 | Column3 | Column 4 |
| 101 | 2001 | 3020 | 01-01-11 |
| 101 | 2001 | 3002 | 02-01-11 |
| 101 | 2001 | 3001 | 03-01-11 |
| 102 | 2002 | 3002 | 01-01-11 |

|  |  |  |  |
| --- | --- | --- | --- |
| 101 | 200131 | 01-01-11 | Site |
| 101 | 200132 | 01-01-11 | Site |
| 101 | 200133 | 01-01-11 | Site |

**select** min(T1.Column1), T2.Column3,

min(T1.Column4), min(T3.Column2)

**from** T1, T2, T3

**where** T1.Column2 = T2.Column2

and T2.Column1 = T3.Column1

**group by** T2.Column3

|  |  |  |
| --- | --- | --- |
| Column1 | Column2 | Column 3 |
| 20011 | 2001 | 200131 |
| 20012 | 2001 | 200132 |
| 20013 | 2001 | 200133 |

|  |  |
| --- | --- |
| Column1 | Column 2 |
| 20011 | Site |
| 20012 | Site |
| 20013 | Site |

**(c)** The output table

**(b)** A SQL query inferred by SQLSythensizer

**(a)** Three input tables: T1 (top), T2 (left), and T3 (right)

|  |  |  |
| --- | --- | --- |
| student\_id | name | level |
| 1 | Adam | senior |
| 2 | Bob | junior |
| 3 | Erin | senior |
| 4 | Rob | junior |
| 5 | Dan | senior |
| 6 | Peter | senior |
| 7 | Sai | senior |

|  |  |  |
| --- | --- | --- |
| student\_id | course\_id | score |
| 1 | 1 | 4 |
| 1 | 2 | 2 |
| 2 | 1 | 3 |
| 2 | 2 | 2 |
| 2 | 3 | 3 |
| 3 | 2 | 1 |
| 4 | 1 | 4 |
| 4 | 3 | 4 |
| 5 | 2 | 5 |
| 5 | 3 | 2 |
| 5 | 4 | 1 |
| 6 | 2 | 4 |
| 6 | 4 | 5 |
| 7 | 1 | 2 |
| 7 | 3 | 3 |
| 7 | 4 | 4 |

|  |  |
| --- | --- |
| name | max\_score |
| Dan | 5 |
| Sai | 5 |

**select** student.name, max(enrolled.score)

**from** student, enrolled

**where** student.student\_id = enrolled.student\_id

and student.level = ‘senior’

**group by** student.student\_id

**having** count(enrolled.course\_id) > 3

**(b)** A SQL query inferred by SQLSynthesizer

**(c)** The output table

**(a)** Two input tables: student (Left) and enrolled (Right)