**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**

**Join conditions**

**Query tables**

**ORDER BY clause**

**HAVING clause**

**GROUP BY clause**

**Query conditions**

**Aggregate**

**Projection column**

**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) > 2

**ORDER BY** student.name

|  |  |
| --- | --- |
| **SELECT** | student.name, **<Aggregate>** |
| **FROM** | student, enrolled |
| **WHERE** | student.student\_id = enrolled.student\_id  and **<Query Condition>** |
| **GROUP BY** | **< Column Name(s)>** |
| **HAVING** | **<Query Condition>** |
| **ORDER BY** | **<Column Name(s)>** |

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

Fgg

ff

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

1. **select** name **from** student **where** score > 2

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

or name = ‘Dan’

**(b)** The output table

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

**Query**

**Skeletons**

**Expected**

**Queries**

**Select**

**Query**

**Candidates**

**Ranked**

**Queries**

**Query**

**Completion**

**Candidate**

**Ranking**

**Input-Output**

**Examples**

**Skeleton Creation**

**Provide more input-output examples**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  |  |  |  |  |  | Group by student\_id  **….** | |
| student\_id | **course\_id** | **score** | **name** | **level** |  | COUNT(**course\_id**) | MAX(**score**) |
| 1 | 1 | 4 | Adam | senior |  | 2 | 4 |
| 1 | 2 | 2 | Adam | senior |  | 2 | 4  **COUNT(course\_id) >2** |
| 2 | 1 | 3 | Bob | junior |  | 3 | 3 |
| 2 | 2 | 2 | Bob | junior |  | 3 | 3  **&& level = “senior”** |
| 2 | 3 | 3 | Bob | junior |  | 3 | 3 |
| 3 | 2 | 1 | Erin | senior |  | 1 | 1 |
| 4 | 1 | 4 | Rob | junior |  | 2 | 4 |
| 4 | 3 | 4 | Rob | junior |  | 2 | 4 |
| 5 | 2 | 5 | Dan | senior |  | 3 | 5 |
| 5 | 3 | 2 | Dan | senior |  | 3 | 5 |
| 5 | 4 | 1 | Dan | senior |  | 3 | 5 |
| 6 | 2 | 4 | Peter | senior |  | 2 | 5 |
| 6 | 4 | 5 | Peter | senior |  | 2 | 5 |
| 7 | 1 | 2 | Sai | senior |  | 3 | 4 |
| 7 | 3 | 3 | Sai | senior |  | 3 | 4 |
| 7 | 4 | 5 | Sai | senior |  | 3 | 4 |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | 5 | Sai | senior |

Aggregation Features

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| 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 | 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 |

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

**Project tuples on**

**column: name, and aggregate: MAX(score)**

**(c)**

**(b)**

**(e)**

**(d)**

**(a)**

**group by student\_id**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| An input table | |  | Aggregation Features | | | | | | | | | | | |  | Comparison Features | | |
| Group by **C1** | | | | | | Group by **C2** | | | | | |
| C1 | C2 | COUNT(**C2**) | COUNT  (DISTINCT **C2**) | MIN(**C2**) | MAX(**C2**) | SUM(**C2**) | AVG(**C2**) | COUNT(**C1**) | COUNT  (DISTINCT **C1**) | MIN(**C1**) | MAX(**C1**) | SUM(**C1**) | AVG(**C1**) | C1 = C2 | C1 < C2 | C1 > C2 |
| 2 | 4 | 3 | 2 | 1 | 4 | 6 | 2 | 1 | 1 | 2 | 2 | 2 | 2 | 0 | 1 | 0 |
| 2 | 1 | 3 | 2 | 1 | 4 | 6 | 2 | 3 | 2 | 1 | 2 | 5 | 5/3 | 0 | 0 | 1 |
| 2 | 1 | 3 | 2 | 1 | 4 | 6 | 2 | 3 | 2 | 1 | 2 | 5 | 5/3 | 0 | 0 | 1 |
| 1 | 1 |  | 1 | 1 | 1 | 1 | 1 | 1 | 3 | 2 | 1 | 2 | 5 | 5/3 |  | 1 | 0 | 0 |

|  |  |  |  |
| --- | --- | --- | --- |
| 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 | 5 |

|  |  |
| --- | --- |
| 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) > 2

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

**(c)** An output table

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