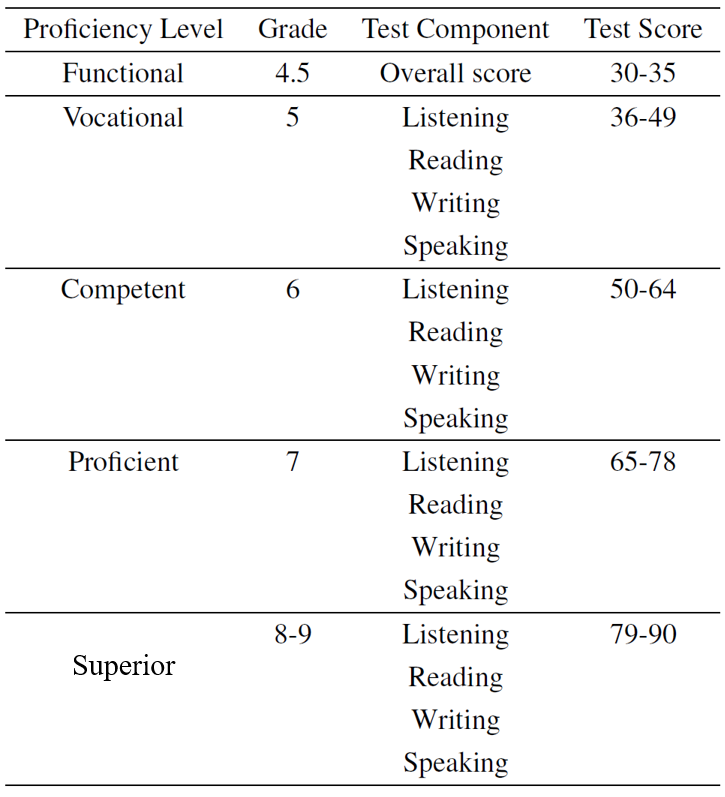
**Tutorial 7**

**PTE Academic Test Case Study**

# Description

PTE Academic Test (administered by Pearson Inc.) is the world’s leading computer-based test of English for study abroad and immigration, which is recognized by universities worldwide, and is approved by the Australian Government for visa applications. PTE Test consists of four main components: Listening, Reading, Speaking, and Writing. A student taking a PTE test will have one score for each of these four components, as well as one overall score. The score is a numerical score, ranging from 10 to 90. The Australian Immigration Department use PTE score to identify an application’s English level of proficiency. The band-scale is shown in Table 1.



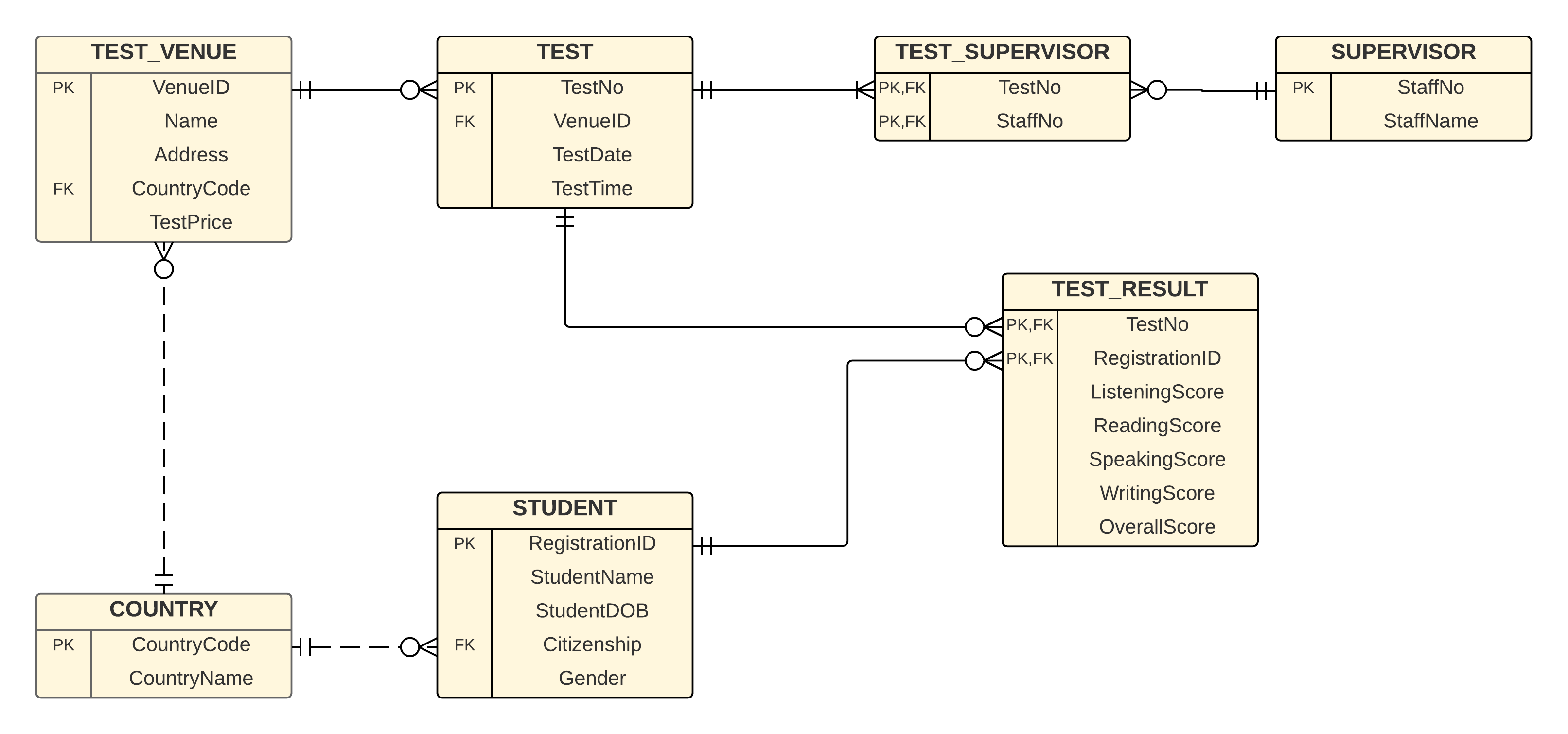
**Table 1:** PTE band-scale

PTE Academic Test has many test venues (all major cities in Australia, and overseas as well). Each country may have a different price setting. For example, taking a PTE test in China is cheaper than taking the same test in Melbourne. PTE Academic Test is a computer-based test. During the test, there is one or more supervisors supervising the test. PTE Academic Test results will normally be released five days after the actual test date. A typical score card contains the student’s details, and the five test scores: Listening, Reading, Writing, Speaking, and Overall.

Generally, universities and the Australian Government look at each component score, rather than the overall score. For example, for an applicant to be able to apply for a Permanent Residence in Australia, he/she must have at least a score of 65 in all of the four components (e.g. proficient in all components: listening, reading, writing, and speaking), regardless his/her overall score.

It is quite common that a student takes a PTE Academic Test several times, in order to improve the score. A student is identified by his/her RegistrationID. The RegistrationID for the same student stays the same and is not changed every time this student sits for a test.

The E/R diagram of the operational system is shown in Figure 1.



**Figure 1:** PTE Academic Test E/R Diagram

You are required to build a data warehouse for this PTE Academic Test system. The data warehouse must be able to answer at least the following questions:

* How many students received a Competent grade in their overall score?
* How many students took the test in 2017?
* How many Korean citizen students took test?
* How many students took the test in Australia?
* How many Chinese students received a Proficient Grade in the Listening part in 2017?

# Task A

Given the requirements above, complete the following:

1. Create a star schema for the PTE Academic Test.
2. Define the dimensions and attributes for the PTE Academic Test star schema.
3. Write the SQL statements for the implementation of the star schema.

The operational database tables can be copied from the **ptetest** account, using:

Select \* from **ptetest**.<table\_name>;

Or

Create Table <your\_table\_name> As

Select …

From **ptetest**.<table\_name>

Where …

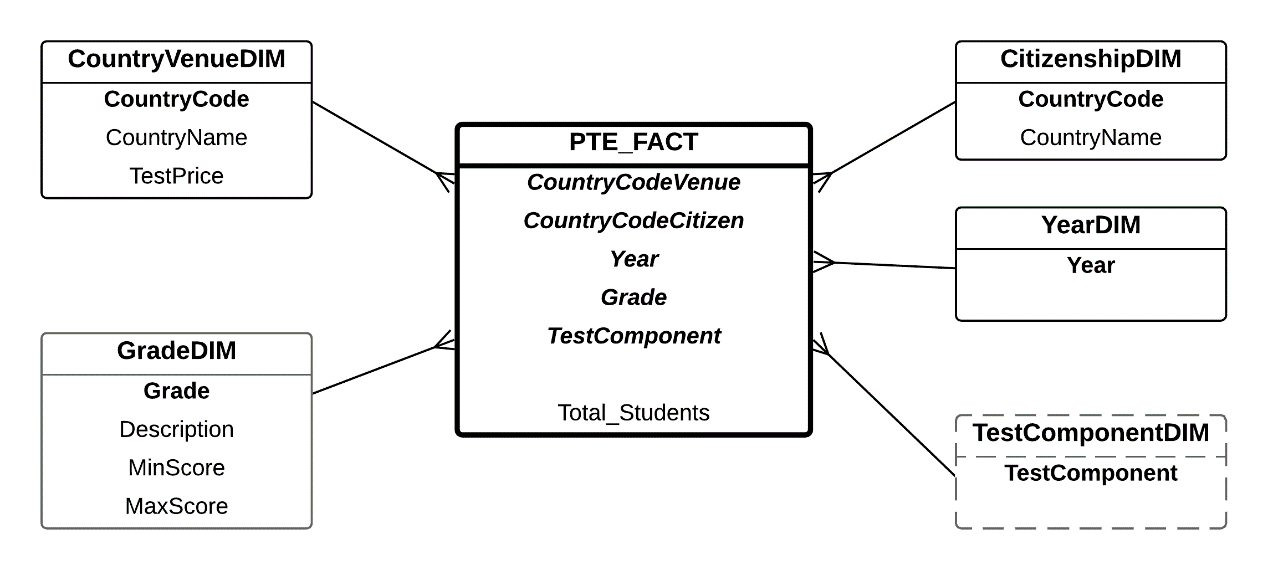
1. Write the SQL statements to produce the following reports:
   1. How many students received a Competent grade in their overall score?
   2. How many students took the test in 2017?
   3. How many Korean citizen students took test?
   4. How many students took the test in Australia?
   5. How many Chinese students received a Proficient Grade in the Listening part in 2017?

# Task B:

1. Design another star schema for the PTE Academic Test that uses **Pivoted Fact Table**.
2. Write the SQL statements for the implementation of the star schema with the Pivoted Fact Table.
3. Write the SQL statements to produce the following reports:
   1. How many students received a Competent grade in their overall score?
   2. How many students took the test in 2017?
   3. How many Korean citizen students took test?
   4. How many students took the test in Australia?
   5. How many Chinese students received a Proficient Grade in the Listening part in 2017?
   6. How many Japanese students received a Competent Grade in 2017?
4. Based on both Task A and Task B, compare the star schema with Determinant Dimension vs. star schema with Pivoted Fact Table.

# Solutions

# Task A: Q1 and Q2: Star Schema



# Task A: Q3. Star Schema Implementation

----------------------------------------------------

-- Step 1: create the dimensions

----------------------------------------------------

-- create Country Venue Dimension

create table CountryVenueDim as

select distinct C.CountryCode, C.CountryName, T.TestPrice

from PTETEST.Test\_Venue T, PTETEST.Country C

where T.CountryCode = C.CountryCode;

-- Note: Country Venue Dimension can be created by joining two tables from the operational database:

-- Test Venue and Country, in order to get the Country Code, Country Name, and Test Price attributes.

-- create Citizenship Dimension

create table CitizenshipDim as

select distinct

C.CountryCode as Citizenship,

C.CountryName

from PTETEST.Student S, PTETEST.Country C

where S.Citizenship = C.CountryCode;

-- Note: For the Citizenship Dimension, it is also a join between Student and Country tables from the operational database, in order to get the Citizenship (which is the Country Code), and Country Name.

-- create Year Dimension

create table YearDim as

select distinct to\_char(TestDate, 'YYYY') as Year

from PTETEST.Test;

-- Note: For the Year dimension, it is basically an extraction from the Test Date attribute in the Test table.

-- create Grade Dimension

create table GradeDim

(Grade varchar2(3),

Description varchar2(20),

MinScore number,

MaxScore number);

-- populate Grade Dimension's data

insert into GradeDim values ('4.5', 'Functional', 30, 35);

insert into GradeDim values ('5', 'Vocational', 36, 49);

insert into GradeDim values ('6', 'Competent', 50, 64);

insert into GradeDim values ('7', 'Proficient', 65, 78);

insert into GradeDim values ('8-9', 'Superior', 79, 90);

-- create Test Component Dimension

create table TestComponentDim

(TestComponent varchar2(20));

-- populate Test Component Dimension's data

insert into TestComponentDim values ('Listening');

insert into TestComponentDim values ('Reading');

insert into TestComponentDim values ('Writing');

insert into TestComponentDim values ('Speaking');

insert into TestComponentDim values ('Overall');

-- Note: For the Grade and Test Component Dimensions, they cannot be extracted from the operational database; they must be created manually.

-- For simplicity, the Test Component Dimension includes only one attribute.

----------------------------------------------------

-- Step 2: create a temp fact table

----------------------------------------------------

-- When a dimension is created manually (instead of extracted from the operational database), we need to create a temporary fact table, called TempFact. In this TempFact, we will get not only Country Code from Test Venue, Citizenship from Student, and TestDate from Test tables, but also the five test scores (e.g. Listening, Reading, Writing, Speaking, and Overall scores), as well as the Student’s RegistrationID.

-- create Temporary Fact table

create table TempFact as

select

TV.CountryCode,

S.Citizenship,

to\_char(T.TestDate, 'YYYY') as Year,

TR.ListeningScore,

TR.ReadingScore,

TR.WritingScore,

TR.SpeakingScore,

TR.OverallScore,

TR.RegistrationID

from PTETEST.Test\_Venue TV, PTETEST.Test T, PTETEST.Student S, PTETEST.Test\_Result TR

where TV.VenueID = T.VenueID

and T.TestNo = TR.TestNo

and TR.RegistrationID = S.RegistrationID;

-- add columns in the tempfact table to store each grade component

alter table TempFact

add (GradeOverall varchar2(3),

GradeListening varchar2(3),

GradeReading varchar2(3),

GradeWriting varchar2(3),

GradeSpeaking varchar2(3)

);

-- converting each score (e.g. Listening score, etc) to a grade

-- the grade is based on the band scale in Table 1

update TempFact

set GradeOverall =

(case

when OverallScore >= 30 and OverallScore <= 35 then '4.5'

when OverallScore >= 36 and OverallScore <= 49 then '5'

when OverallScore >= 50 and OverallScore <= 64 then '6'

when OverallScore >= 65 and OverallScore <= 78 then '7'

when OverallScore >= 79 and OverallScore <= 90 then '8-9'

end);

update TempFact set GradeListening =

(case

when ListeningScore >= 30 and ListeningScore <= 35 then '4.5'

when ListeningScore >= 36 and ListeningScore <= 49 then '5'

when ListeningScore >= 50 and ListeningScore <= 64 then '6'

when ListeningScore >= 65 and ListeningScore <= 78 then '7'

when ListeningScore >= 79 and ListeningScore <= 90 then '8-9'

end);

update TempFact set GradeReading =

(case

when ReadingScore >= 30 and ReadingScore <= 35 then '4.5'

when ReadingScore >= 36 and ReadingScore <= 49 then '5'

when ReadingScore >= 50 and ReadingScore <= 64 then '6'

when ReadingScore >= 65 and ReadingScore <= 78 then '7'

when ReadingScore >= 79 and ReadingScore <= 90 then '8-9'

end);

update TempFact set GradeWriting =

(case

when WritingScore >= 30 and WritingScore <= 35 then '4.5'

when WritingScore >= 36 and WritingScore <= 49 then '5'

when WritingScore >= 50 and WritingScore <= 64 then '6'

when WritingScore >= 65 and WritingScore <= 78 then '7'

when WritingScore >= 79 and WritingScore <= 90 then '8-9'

end);

update TempFact set GradeSpeaking =

(case

when SpeakingScore >= 30 and SpeakingScore <= 35 then '4.5'

when SpeakingScore >= 36 and SpeakingScore <= 49 then '5'

when SpeakingScore >= 50 and SpeakingScore <= 64 then '6'

when SpeakingScore >= 65 and SpeakingScore <= 78 then '7'

when SpeakingScore >= 79 and SpeakingScore <= 90 then '8-9'

end);

----------------------------------------------------

-- Step 3: create temporary fact table

-- for each test component

----------------------------------------------------

-- Note: After the updates are completed, the TempFact table has the correct grades for each student. The problem now is to breakdown the grades for a student into multiple test components. This can be done by creating yet another temporary fact for each of the test components. For example, for the Overall Score, we need a temporary fact, called ’OverallFact’, and for the Listening Score, we will have ’ListeningFact’. In other words, we will have five temporary fact tables – one for each test component. The structure of the OverallFact, for example, is Country Code, Citizenship, Year, GradeOverall (which we got from the above update statement), the string ’Overall’ which represents the Test Component, and count of RegistrationID for Total Students. This process is repeated for the other four temporary facts, called ListeningFact, ReadingFact, WritingFact, and SpeakingFact.

create table OverallFact as

select

CountryCode,

Citizenship,

Year,

GradeOverall As Grade,

'Overall' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeOverall,

'Overall';

create table ListeningFact as

select

CountryCode,

Citizenship,

Year,

GradeListening As Grade,

'Listening' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeListening,

'Listening';

create table ReadingFact as

select

CountryCode,

Citizenship,

Year,

GradeReading As Grade,

'Reading' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeReading,

'Reading';

create table WritingFact as

select

CountryCode,

Citizenship,

Year,

GradeWriting As Grade,

'Writing' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeWriting,

'Writing';

create table SpeakingFact as

select

CountryCode,

Citizenship,

Year,

GradeSpeaking As Grade,

'Speaking' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeSpeaking,

'Speaking';

----------------------------------------------------

-- Step 4: create the final fact table

----------------------------------------------------

-- Note: After creating the five temporary facts, these fact tables can now be combined to form one final fact table. The final fact table will be the union of the five fact tables.

create table FinalFact as

select

CountryCode,

Citizenship,

Year,

Grade,

TestComponent,

Total\_Students\_Overall as Total\_Students

from OverallFact

union

select \* from ListeningFact

union

select \* from ReadingFact

union

select \* from WritingFact

union

select \* from SpeakingFact;

-- Note: Note that the column names of the FinalFact will be based on the first fact in the Union, which is the OverallFact. The FinalFact table itself contains 45 records, which is the combination of the five fact tables.

# Task A: Q4. The Reports

a) How many students received a Competent grade in their overall score?

SELECT g.grade, g.description as grade\_description, t.testcomponent,

SUM(f.total\_students) as number\_of\_students

FROM FinalFact f, TestComponentDim t, GradeDim g

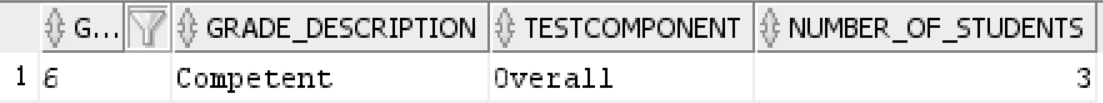
WHERE f.grade = g.grade

AND f.testcomponent = t.testcomponent

AND g.description = 'Competent'

AND t.testcomponent = 'Overall'

GROUP BY g.grade, g.description, t.testcomponent;



1 row selected.

b) How many students took the test in 2017?

SELECT y.year, t.testcomponent, SUM(f.total\_students) as number\_of\_students

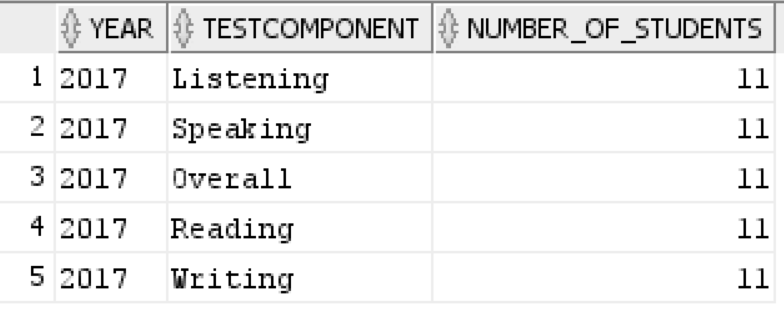
FROM FinalFact f, TestComponentDim t, YearDim y

WHERE f.year = y.year

AND f.testcomponent = t.testcomponent

AND y.year = '2017'

GROUP BY y.year, t.testcomponent;



5 rows selected.

**Note:** The answer below is **INCORRECT**. If you check the operational database (specifically in the test\_result table), there are only 11 records of students taking the test in 2017. If you do not take into consideration of the test components when querying the data warehouse, the result of the query is incorrect as it will sum up all the number of students of all components as shown in the result below.

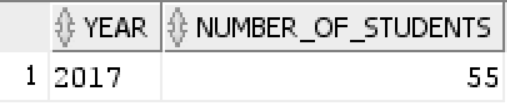
SELECT y.year, SUM(f.total\_students) as number\_of\_students

FROM FinalFact f, YearDim y

WHERE f.year = y.year

AND y.year = '2017'

GROUP BY y.year;



1 row selected.

c) How many Korean citizen students took test?

SELECT c.countryname, t.testcomponent,

SUM(f.total\_students) as number\_of\_students

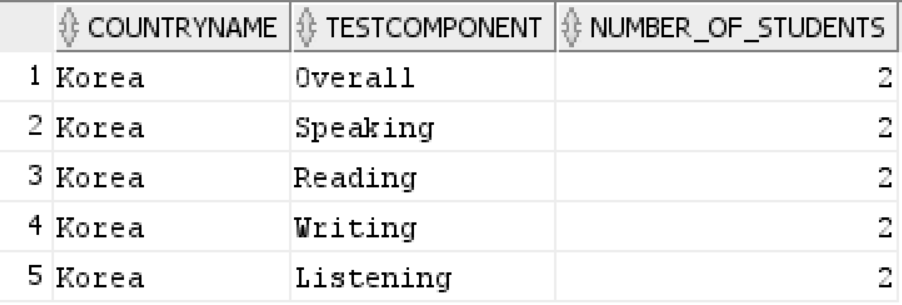
FROM FinalFact f, TestComponentDim t, CitizenshipDim c

WHERE f.citizenship = c.citizenship

AND f.testcomponent = t.testcomponent

AND c.countryname = 'Korea'

GROUP BY c.countryname, t.testcomponent;



5 rows selected.

d) How many students took the test in Australia?

SELECT c.countrycode, c.countryname, t.testcomponent,

SUM(f.total\_students) as number\_of\_students

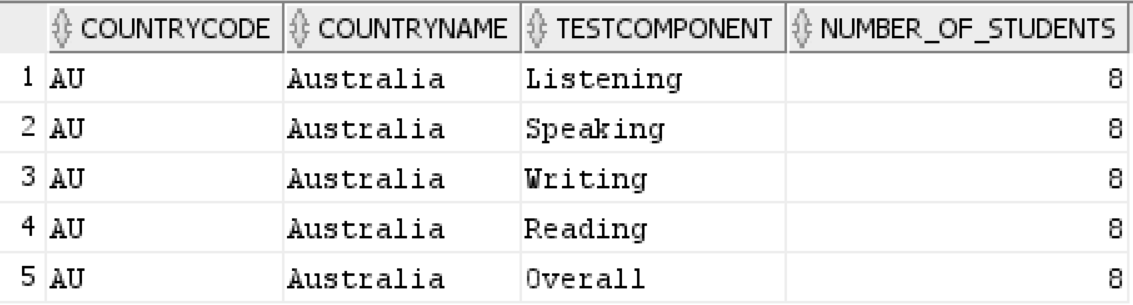
FROM FinalFact f, TestComponentDim t, CountryVenueDim c

WHERE f.countrycode = c.countrycode

AND f.testcomponent = t.testcomponent

AND c.countryname = 'Australia'

GROUP BY c.countrycode, c.countryname, t.testcomponent;



5 rows selected.

e) How many Chinese students received a Proficient Grade in the Listening part in 2017?

SELECT c.countryname as citizenship,

g.grade,

g.description AS grade\_description,

t.testcomponent,

y.year,

SUM(f.total\_students) as number\_of\_students

FROM FinalFact f,

TestComponentDim t,

CitizenshipDim c,

GradeDim g,

YearDim y

WHERE f.testcomponent = t.testcomponent

AND f.citizenship = c.citizenship

AND f.grade = g.grade

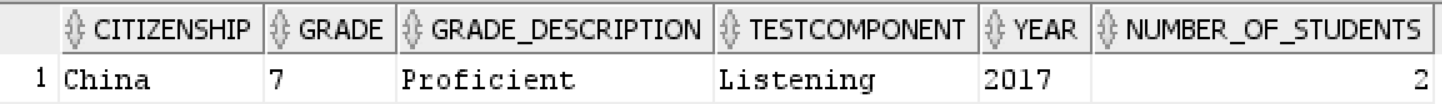
AND f.year = y.year

AND c.countryname = 'China'

AND g.description = 'Proficient'

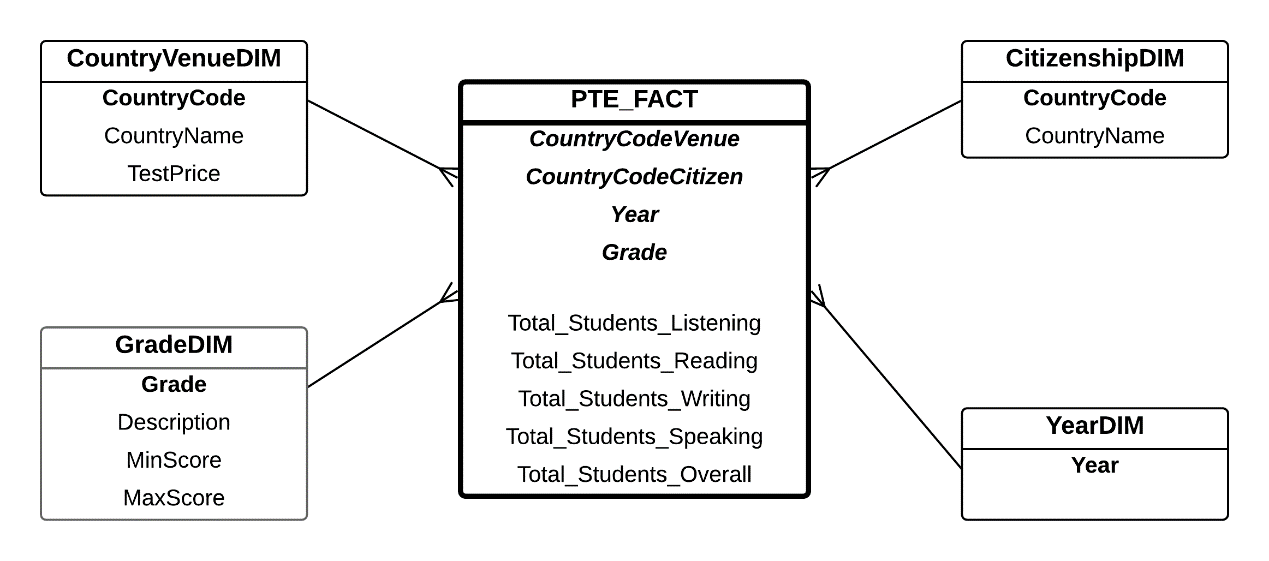
AND t.testcomponent = 'Listening'

GROUP BY c.countryname, g.grade, g.description, t.testcomponent, y.year;



1 row selected.

# Task B: Q1. Star Schema – Pivoted Fact Table Version



In this star schema, the Test Component dimension is removed, but there is a fact measure for each of the test components (e.g. Total Students Listening, Total Student Reading, etc). This means that the fact table is a ***Pivoted Fact Table***, in which for each record in the fact table, it can be seen as a 2D matrix between the dimension key identifiers and the five fact measures.

# Task B: Q2. Star Schema Implementation

----------------------------------------------------

-- Step 1: create the dimensions

----------------------------------------------------

-- Note: The four dimension tables that have been created in the previous section can be re-used here.

-- create Country Venue Dimension

create table CountryVenueDim as

select distinct C.CountryCode, C.CountryName, T.TestPrice

from PTETEST.Test\_Venue T, PTETEST.Country C

where T.CountryCode = C.CountryCode;

-- Country Venue Dimension can be created by joining two tables from the operational database:

-- Test Venue and Country, in order to get the Country Code, Country Name, and Test Price attributes.

-- create Citizenship Dimension

create table CitizenshipDim as

select distinct

C.CountryCode as Citizenship,

C.CountryName

from PTETEST.Student S, PTETEST.Country C

where S.Citizenship = C.CountryCode;

-- For the Citizenship Dimension, it is also a join between Student and Country tables from the operational database, in order to get the Citizenship (which is the Country Code), and Country Name.

-- create Year Dimension

create table YearDim as

select distinct to\_char(TestDate, 'YYYY') as Year

from PTETEST.Test;

-- For the Year dimension, it is basically an extraction from the Test Date attribute in the Test table.

-- create Grade Dimension

create table GradeDim

(Grade varchar2(3),

Description varchar2(20),

MinScore number,

MaxScore number);

-- populate Grade Dimension's data

insert into GradeDim values ('4.5', 'Functional', 30, 35);

insert into GradeDim values ('5', 'Vocational', 36, 49);

insert into GradeDim values ('6', 'Competent', 50, 64);

insert into GradeDim values ('7', 'Proficient', 65, 78);

insert into GradeDim values ('8-9', 'Superior', 79, 90);

-- The Grade Dimensions cannot be extracted from the operational database, thus it must be created manually.

----------------------------------------------------

-- Step 2: create a temp fact table

----------------------------------------------------

-- Note: This step is similar to the step in Task A.

-- create Temporary Fact table

create table TempFact as

select

TV.CountryCode,

S.Citizenship,

to\_char(T.TestDate, 'YYYY') as Year,

TR.ListeningScore,

TR.ReadingScore,

TR.WritingScore,

TR.SpeakingScore,

TR.OverallScore,

TR.RegistrationID

from PTETEST.Test\_Venue TV, PTETEST.Test T, PTETEST.Student S, PTETEST.Test\_Result TR

where TV.VenueID = T.VenueID

and T.TestNo = TR.TestNo

and TR.RegistrationID = S.RegistrationID;

-- add columns in the tempfact table to store each grade component

alter table TempFact

add (GradeOverall varchar2(3),

GradeListening varchar2(3),

GradeReading varchar2(3),

GradeWriting varchar2(3),

GradeSpeaking varchar2(3)

);

-- converting each score (e.g. Listening score, etc) to a grade

-- the grade is based on the band scale in Table 1

update TempFact

set GradeOverall =

(case

when OverallScore >= 30 and OverallScore <= 35 then '4.5'

when OverallScore >= 36 and OverallScore <= 49 then '5'

when OverallScore >= 50 and OverallScore <= 64 then '6'

when OverallScore >= 65 and OverallScore <= 78 then '7'

when OverallScore >= 79 and OverallScore <= 90 then '8-9'

end);

update TempFact set GradeListening =

(case

when ListeningScore >= 30 and ListeningScore <= 35 then '4.5'

when ListeningScore >= 36 and ListeningScore <= 49 then '5'

when ListeningScore >= 50 and ListeningScore <= 64 then '6'

when ListeningScore >= 65 and ListeningScore <= 78 then '7'

when ListeningScore >= 79 and ListeningScore <= 90 then '8-9'

end);

update TempFact set GradeReading =

(case

when ReadingScore >= 30 and ReadingScore <= 35 then '4.5'

when ReadingScore >= 36 and ReadingScore <= 49 then '5'

when ReadingScore >= 50 and ReadingScore <= 64 then '6'

when ReadingScore >= 65 and ReadingScore <= 78 then '7'

when ReadingScore >= 79 and ReadingScore <= 90 then '8-9'

end);

update TempFact set GradeWriting =

(case

when WritingScore >= 30 and WritingScore <= 35 then '4.5'

when WritingScore >= 36 and WritingScore <= 49 then '5'

when WritingScore >= 50 and WritingScore <= 64 then '6'

when WritingScore >= 65 and WritingScore <= 78 then '7'

when WritingScore >= 79 and WritingScore <= 90 then '8-9'

end);

update TempFact set GradeSpeaking =

(case

when SpeakingScore >= 30 and SpeakingScore <= 35 then '4.5'

when SpeakingScore >= 36 and SpeakingScore <= 49 then '5'

when SpeakingScore >= 50 and SpeakingScore <= 64 then '6'

when SpeakingScore >= 65 and SpeakingScore <= 78 then '7'

when SpeakingScore >= 79 and SpeakingScore <= 90 then '8-9'

end);

----------------------------------------------------

-- Step 3: create temporary fact table

-- for each test component

----------------------------------------------------

-- Note: This step is similar to the step in Task A as we will need these five temporary fact tables to create the Pivoted Fact Table.

create table OverallFact as

select

CountryCode,

Citizenship,

Year,

GradeOverall As Grade,

'Overall' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeOverall,

'Overall';

create table ListeningFact as

select

CountryCode,

Citizenship,

Year,

GradeListening As Grade,

'Listening' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeListening,

'Listening';

create table ReadingFact as

select

CountryCode,

Citizenship,

Year,

GradeReading As Grade,

'Reading' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeReading,

'Reading';

create table WritingFact as

select

CountryCode,

Citizenship,

Year,

GradeWriting As Grade,

'Writing' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeWriting,

'Writing';

create table SpeakingFact as

select

CountryCode,

Citizenship,

Year,

GradeSpeaking As Grade,

'Speaking' as TestComponent,

count(RegistrationID) as Total\_Students\_Overall

from TempFact

group by

CountryCode,

Citizenship,

Year,

GradeSpeaking,

'Speaking';

----------------------------------------------------

-- Step 4: create a Cartesian Product of all

-- dimensions in order to get all possible

-- combinations of the dimensions

----------------------------------------------------

-- **NOTE:** In general, when a fact table is created, it may not contain all possible combinations from all records of all dimensions. The reason is that the fact table is created by a join (an inner join) operation. Consequently, combinations of records from the dimension tables that do not have any value for the fact measure will not be included in the fact table – simply put, there is no zero values in the fact measure.

-- If we create a fact table as in the star schema with Pivoted Fact Table that has five fact measures: Total Students with Listening, with Reading, with Writing, with Speaking, and with Overall Score; it is expected that we should have 60 records in the final fact table (2 countries of test venues, 6 countries of citizenship, 1 year, and 5 grades) and we also have to keep track of the zero values.

-- In order to achieve this, first we need to get all possible combinations from all dimensions, which is a Cartesian Product between all dimensions.

create table AllDimensions as

select

CO.CountryCode,

CI.Citizenship,

Y.Year,

G.Grade

from

CountryVenueDim CO,

CitizenshipDim CI,

YearDim Y,

GradeDim G;

----------------------------------------------------

-- Step 5: create temporary fact table

-- for each test component

----------------------------------------------------

-- **NOTE:** Next is to re-create five temporary fact tables, one for each test component. These temporary fact tables are created by an *Outer Join* operation between AllDimensions table and each of the temporary fact tables created in the previous section. An outer join operation is used so that we preserve all records from the AllDimensions table. The Total Student will be zero when the record in the AllDimensions table does not match with any of the records from the temporary fact (Note: an nvl function is used).

create table OverallFactNew as

select

A.CountryCode,

A.Citizenship,

A.Year,

A.Grade,

nvl(O.Total\_Students\_Overall, 0)

as Total\_Students\_Overall

from AllDimensions A, OverallFact O

where A.CountryCode = O.CountryCode(+)

and A.Citizenship = O.Citizenship(+)

and A.Year = O.Year(+)

and A.Grade = O.Grade(+);

create table ListeningFactNew as

select

A.CountryCode,

A.Citizenship,

A.Year,

A.Grade,

nvl(O.Total\_Students\_Overall, 0)

as Total\_Students\_Listening

from AllDimensions A, ListeningFact O

where A.CountryCode = O.CountryCode(+)

and A.Citizenship = O.Citizenship(+)

and A.Year = O.Year(+)

and A.Grade = O.Grade(+);

create table ReadingFactNew as

select

A.CountryCode,

A.Citizenship,

A.Year,

A.Grade,

nvl(O.Total\_Students\_Overall, 0)

as Total\_Students\_Reading

from AllDimensions A, ReadingFact O

where A.CountryCode = O.CountryCode(+)

and A.Citizenship = O.Citizenship(+)

and A.Year = O.Year(+)

and A.Grade = O.Grade(+);

create table WritingFactNew as

select

A.CountryCode,

A.Citizenship,

A.Year,

A.Grade,

nvl(O.Total\_Students\_Overall, 0)

as Total\_Students\_Writing

from AllDimensions A, WritingFact O

where A.CountryCode = O.CountryCode(+)

and A.Citizenship = O.Citizenship(+)

and A.Year = O.Year(+)

and A.Grade = O.Grade(+);

create table SpeakingFactNew as

select

A.CountryCode,

A.Citizenship,

A.Year,

A.Grade,

nvl(O.Total\_Students\_Overall, 0)

as Total\_Students\_Speaking

from AllDimensions A, SpeakingFact O

where A.CountryCode = O.CountryCode(+)

and A.Citizenship = O.Citizenship(+)

and A.Year = O.Year(+)

and A.Grade = O.Grade(+);

-- **NOTE:** Once the five temporary fact tables are created, the final fact table can be created. Note that the structure of the five temporary tables is identical, that is Country Code, Citizenship, Year, Grade, and Total Students – there is no Test Component. The Total Students column is basically the total students for the particular temporary fact; that means for the ReadingFactNew table, the Total Students column is the Total Students for the Reading Test Component – that’s why we have five temporary fact tables; one

for each test component.

----------------------------------------------------

-- Step 6: create the final fact table

----------------------------------------------------

-- **NOTE:** In order to create the final fact table, we use the join operation. Remember that for the determinant version of the fact table, a union operator was used. But for the non-determinant version, here we use a join operator to join the five temporary facts based on Country, Citizenship, Year, and Grade. We will still keep each Total Student column from each of the five temporary fact.

create table FinalFact2 as

select

O.CountryCode,

O.Citizenship,

O.Year,

O.Grade,

O.Total\_Students\_Overall,

L.Total\_Students\_Listening,

R.Total\_Students\_Reading,

W.Total\_Students\_Writing,

S.Total\_Students\_Speaking

from

OverallFactNew O,

ListeningFactNew L,

ReadingFactNew R,

WritingFactNew W,

SpeakingFactNew S

where O.CountryCode = L.CountryCode

and L.CountryCode = R.CountryCode

and R.CountryCode = W.CountryCode

and W.CountryCode = S.CountryCode

and O.Citizenship = L.Citizenship

and L.Citizenship = R.Citizenship

and R.Citizenship = W.Citizenship

and W.Citizenship = S.Citizenship

and O.Year = L.Year

and L.Year = R.Year

and R.Year = W.Year

and W.Year = S.Year

and O.Grade = L.Grade

and L.Grade = R.Grade

and R.Grade = W.Grade

and W.Grade = S.Grade;

----------------------------------------------------

-- Step 7: delete unnecessary data in

-- the final fact table

----------------------------------------------------

-- **NOTE:** Because the data in the operational database is small, there are many records with zeroes values in the Total Students columns. In order to exclude those records, we can simply delete records where Total Students in all of the five test components are equal to zero.

delete from FinalFact2

where Total\_Students\_Overall = 0

and Total\_Students\_Listening = 0

and Total\_Students\_Reading = 0

and Total\_Students\_Writing = 0

and Total\_Students\_Speaking = 0;

-- The Final Fact table only contains eleven records since all records that comprises all zero values for all test components are removed.

# Task B: Q3. The Reports

a) How many students received a Competent grade in their overall score?

SELECT g.grade, g.description as grade\_description,

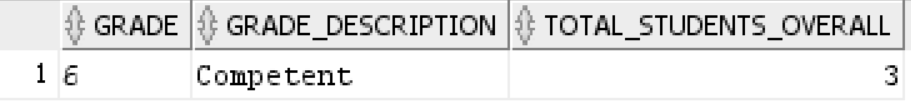
SUM(f.total\_students\_overall) AS total\_students\_overall

FROM FinalFact2 f, GradeDim g

WHERE f.grade = g.grade

AND g.description = 'Competent'

GROUP BY g.grade, g.description;



1 row selected.

b) How many students took the test in 2017?

SELECT y.year,

SUM(f.total\_students\_overall) as total\_students\_overall,

SUM(f.total\_students\_listening) as total\_students\_listening,

SUM(f.total\_students\_reading) as total\_students\_reading,

SUM(f.total\_students\_writing) as total\_students\_writing,

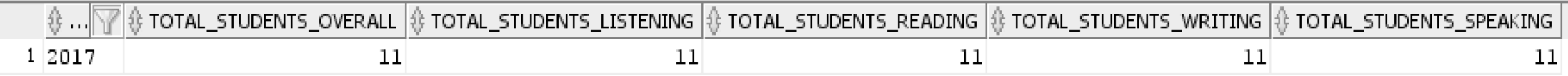
SUM(f.total\_students\_speaking) as total\_students\_speaking

FROM FinalFact2 f, YearDim y

WHERE f.year = y.year

AND y.year = '2017'

GROUP BY y.year;



1 row selected.

c) How many Korean citizen students took test?

SELECT c.countryname,

SUM(f.total\_students\_overall) as total\_students\_overall,

SUM(f.total\_students\_listening) as total\_students\_listening,

SUM(f.total\_students\_reading) as total\_students\_reading,

SUM(f.total\_students\_writing) as total\_students\_writing,

SUM(f.total\_students\_speaking) as total\_students\_speaking

FROM FinalFact2 f, CitizenshipDim c

WHERE f.citizenship = c.citizenship

AND c.countryname = 'Korea'

GROUP BY c.countryname;



1 row selected.

d) How many students took the test in Australia?

SELECT c.countrycode,

c.countryname,

SUM(f.total\_students\_overall) as total\_students\_overall,

SUM(f.total\_students\_listening) as total\_students\_listening,

SUM(f.total\_students\_reading) as total\_students\_reading,

SUM(f.total\_students\_writing) as total\_students\_writing,

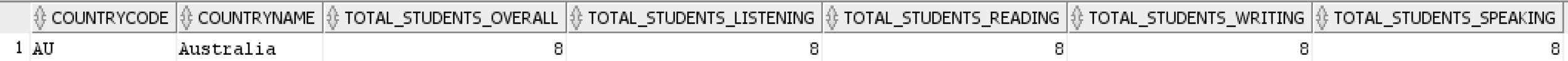
SUM(f.total\_students\_speaking) as total\_students\_speaking

FROM FinalFact2 f, CountryVenueDim c

WHERE f.countrycode = c.countrycode

AND c.countryname = 'Australia'

GROUP BY c.countrycode, c.countryname;



1 row selected.

e) How many Chinese students received a Proficient Grade in the Listening part in 2017?

SELECT c.countryname as citizenship,

g.grade,

g.description AS grade\_description,

y.year,

SUM(f.total\_students\_listening) as total\_students\_listening

FROM FinalFact2 f, CitizenshipDim c, GradeDim g, YearDim y

WHERE f.citizenship = c.citizenship

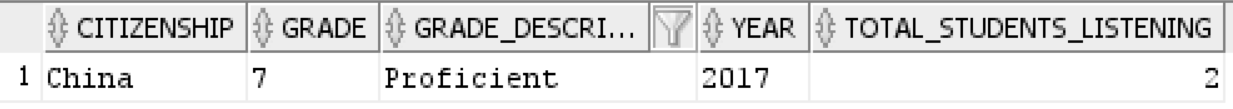
AND f.grade = g.grade

AND f.year = y.year

AND c.countryname = 'China'

AND g.description = 'Proficient'

GROUP BY c.countryname, g.grade, g.description, y.year;



1 row selected.

f) How many Japanese students received a Competent Grade in 2017?

SELECT c.countryname as citizenship,

g.grade,

g.description AS grade\_description,

y.year,

SUM(f.total\_students\_overall) as total\_students\_overall,

SUM(f.total\_students\_listening) as total\_students\_listening,

SUM(f.total\_students\_reading) as total\_students\_reading,

SUM(f.total\_students\_writing) as total\_students\_writing,

SUM(f.total\_students\_speaking) as total\_students\_speaking

FROM FinalFact2 f, CitizenshipDim c, GradeDim g, YearDim y

WHERE f.citizenship = c.citizenship

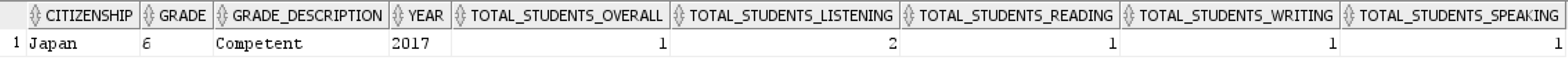
AND f.grade = g.grade

AND f.year = y.year

AND c.countryname = 'Japan'

AND g.description = 'Competent'

GROUP BY c.countryname, g.grade, g.description, y.year;



1 row selected.

# Task B: Q4. Determinant Dimension vs. Pivoted Fact Table

|  |  |  |
| --- | --- | --- |
| Differences | Determinant Dimension | Pivoted Fact Table |
| Storage | More records in the Fact table due to the additional test components | Lesser records but more columns |
| Modelling | More concise and compact with a smaller number of measures in the fact, easier to understand | The schema looks more  complex and crowded; but the storage cost is lower |
| Implementation | Shorter steps compared to the Pivoted Fact Table | Longer process as the creation of the Non-Determinant version (or the Pivoted Fact version) uses the temporary fact tables from the Determinant version |
| Query Processing | Requires additional join with  TestComponentDIM | Less join processing between  dimension tables and the fact  table. |

**THE END**