COMS W4111-002/V002 (Spring 2023) Introduction to Databases

</span>

Homework 4: All Tracks

Overview

- There are two parts to HW 4:
  - `4a` : Written questions
  - `4b:` A common set of practical tasks for both the programming and non-programming tracks.
- HW 4 **does not** have separate assignments for the programming and non-programming tracks.

Homework 4b has the following tasks:

1. Create a new schema `<uni>_S22_classic_models_star.` Replace `<uni>` with your UNI.

2. You will create a star schema using the data from your Classic Models database.
   - The fact in the fact table is of the form `(productCode, quantityPrders, priceEach, orderedData, customerNumber).`
   - The dimensions are:
     - `date_dimension: year, quarter, month, day of the month.`
     - `location_dimension: region, country, city.` The zip file contain a file `country_region.csv` that provides the mapping of countries to regions.
     - `product_dimension: product_scale, product_line, product_vendor.`

3. You will write queries that demonstrate:
   - A slice of the data.
   - A dice of the data.
   - A drill-down.
   - A roll-up.

- The homework is due on `2022-MAY-01` at 11:59 PM. We will post detailed submission instructions on Ed and Gradescope. Your submission format will be PDF and zip copies of this notebook. You must name your files following the instructions we publish.

Setup

In [1]:

```
import pandas as pd
```

In [2]:

```
%load_ext sql
```

In [3]:

```
%sql mysql+pymysql://root:Edy990127@localhost
```

Out[3]:

```
'Connected: root@None'
```

In [4]:

```
country_region = pd.read_csv('./country_region.csv')
```

In [5]:

```
country_region
```

Out[5]:

	Country	Region
0	France	EMEA
1	USA	NaN
2	Australia	APAC
3	Norway	EMEA
4	Poland	EMEA
5	Germany	EMEA
6	Spain	EMEA
7	Sweden	EMEA
8	Denmark	EMEA
9	Singapore	APAC
10	Portugal	EMEA
11	Japan	APAC
12	Finland	EMEA
13	UK	EMEA
14	Ireland	EMEA
15	Canada	NaN
16	Hong Kong	APAC
17	Italy	EMEA
18	Switzerland	EMEA
19	Netherlands	EMEA
20	Belgium	EMEA
21	New Zealand	APAC
22	South Africa	EMEA

	Country	Region
23	Austria	APAC
24	Philippines	APAC
25	Russia	EMEA
26	Israel	EMEA

Schema

- Execute your SQL statements for creating the schema, table and constraints for the fact and dimension tables in the following cells.

In [6]:

```
#%sql create schema de2418_S22_classic_models_star
```

In [7]:

```
%%sql
drop table if exists de2418_S22_classic_models_star.date_dimension;
create table de2418_S22_classic_models_star.date_dimension
(
    year             text  not null,
    orderedDate      varchar(20) default '0' not null
        primary key,
    quarter          text not null,
    month            text not null,
    day_of_the_month text not null
);
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[7]:

```
[]
```

In [8]:

```
%%sql
drop table if exists de2418_S22_classic_models_star.location_dimension;
create table de2418_S22_classic_models_star.location_dimension
(
    customerNumber varchar(20) not null
        primary key,
    city           text        ,
    country        text        ,
    region         text
);
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[8]:

```
[]
```

In [9]:

```
%%sql
drop table if exists de2418_S22_classic_models_star.product_dimension;
```

```
create table de2418_S22_classic_models_star.product_dimension
(
    product_Code   varchar(20) default '0' not null
        primary key,
    product_scale  text not null,
    product_line   text not null,
    product_vendor text not null
);
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[9]:

```
[]
```

In [10]:

```
%%sql
drop table if exists de2418_S22_classic_models_star.fact_table;
create table de2418_S22_classic_models_star.fact_table
(
    productCode     varchar(20)                 not null,
    QuantityOrdered text    not null,
    priceEach       text not null,
    orderedDate     varchar(20)    default '0'  not null,
    customerNumber  varchar(20)    default '0'  not null,
    primary key (productCode, orderedDate, customerNumber),
    constraint fact_table_date_dimension_orderedDate_fk
        foreign key (orderedDate) references date_dimension (orderedDate),
    constraint fact_table_location_dimension_customerNumber_fk
        foreign key (customerNumber) references location_dimension (customerNumber),
    constraint fact_table_product_dimension_product_Code_fk
        foreign key (productCode) references product_dimension (product_Code)
);
```

```
 * mysql+pymysql://root:***@localhost
0 rows affected.
0 rows affected.
```

Out[10]:

```
[]
```

Data Loading

- Enter and execute your SQL for loading the data into the facts and dimensions table. The source of the information is the Classic Models data.

In [11]:

```
from sqlalchemy import create_engine
sql_engine = create_engine('mysql+pymysql://root:Edy990127@localhost')
```

In [12]:

```
country_region.to_sql(
    "country_region", con=sql_engine, if_exists="replace", index=False,
    schema="de2418_S22_classic_models_star")
```

In [13]:

```
%%sql
insert into de2418_S22_classic_models_star.date_dimension (orderedDate, year, quarter,
select a.orderDate, year(a.orderDate), quarter(a.orderDate), month(a.OrderDate), day(a.
from (select distinct orderDate from classicmodels.orders) as a
```

```
 * mysql+pymysql://root:***@localhost
265 rows affected.
```

Out[13]:

```
[]
```

In [14]:

```
%%sql
insert into de2418_S22_classic_models_star.location_dimension (customerNumber, city, co
select a.customerNumber, a.city, b.Country, b.Region from
(select customerNumber, city, country from classicmodels.customers) as a
left join de2418_S22_classic_models_star.country_region as b
on a.country = b.country
```

```
 * mysql+pymysql://root:***@localhost
122 rows affected.
```

Out[14]:

```
[]
```

In [15]:

```
%%sql
insert into de2418_S22_classic_models_star.product_dimension (product_Code, product_sca
select productCode, productScale, productLine, productVendor from classicmodels.product
```

```
 * mysql+pymysql://root:***@localhost
110 rows affected.
```

Out[15]:

```
[]
```

In [16]:

```
%%sql
insert into de2418_S22_classic_models_star.fact_table (productCode, QuantityOrdered, pr
select a.productCode, a.quantityOrdered, a.priceEach, b.orderDate, b.customerNumber fro
(select orderNumber, productCode, quantityOrdered, priceEach from classicmodels.orderde
left join (select orderNumber, orderDate, customerNumber from classicmodels.orders) as
on a.orderNumber = b.orderNumber
```

```
 * mysql+pymysql://root:***@localhost
2996 rows affected.
```

Out[16]:

```
[]
```

Queries

- In each of the sections below, define what your query is producing, provide the query and execute to produce the results.

In [17]:

```
%%sql
create view de2418_S22_classic_models_star.sales_view as
select a.productCode, a.QuantityOrdered, a.priceEach, a.orderedDate, a.customerNumber,
b.product_scale, b.product_line, b.product_vendor,
c.city, c.country, c.region,
d.year, d.quarter, d.month, d.day_of_the_month from
(select * from de2418_S22_classic_models_star.fact_table) as a
join de2418_S22_classic_models_star.product_dimension as b
on a.productCode = b.product_Code
```

```
join de2418_S22_classic_models_star.location_dimension as c
on a.customerNumber = c.customerNumber
join de2418_S22_classic_models_star.date_dimension as d
on a.orderedDate = d.orderedDate
```

```
 * mysql+pymysql://root:***@localhost
(pymysql.err.OperationalError) (1050, "Table 'sales_view' already exists")
[SQL: create view de2418_S22_classic_models_star.sales_view as
select a.productCode, a.QuantityOrdered, a.priceEach, a.orderedDate, a.customerNumber,
b.product_scale, b.product_line, b.product_vendor,
c.city, c.country, c.region,
d.year, d.quarter, d.month, d.day_of_the_month from
(select * from de2418_S22_classic_models_star.fact_table) as a
join de2418_S22_classic_models_star.product_dimension as b
on a.productCode = b.product_Code
join de2418_S22_classic_models_star.location_dimension as c
on a.customerNumber = c.customerNumber
join de2418_S22_classic_models_star.date_dimension as d
on a.orderedDate = d.orderedDate]
(Background on this error at: https://sqlalche.me/e/14/e3q8)
```

In [18]:

```
%sql select * from de2418_S22_classic_models_star.sales_view LIMIT 30
```

```
 * mysql+pymysql://root:***@localhost
30 rows affected.
```

Out[18]:

productCode	QuantityOrdered	priceEach	orderedDate	customerNumber	product_scale	product_line
S10_1678	30	81.35	2003-02-24	131	1:10	Motorcycles
S10_1678	34	86.13	2003-05-07	353	1:10	Motorcycles
S10_1678	41	90.92	2003-07-01	250	1:10	Motorcycles
S10_1678	45	76.56	2003-08-25	205	1:10	Motorcycles
S10_1678	49	81.35	2003-10-10	321	1:10	Motorcycles
S10_1678	36	94.74	2003-10-28	161	1:10	Motorcycles
S10_1678	29	76.56	2003-11-11	171	1:10	Motorcycles
S10_1678	48	95.70	2003-11-18	167	1:10	Motorcycles
S10_1678	22	82.30	2003-12-01	129	1:10	Motorcycles
S10_1678	41	90.92	2004-01-15	406	1:10	Motorcycles
S10_1678	37	80.39	2004-02-20	114	1:10	Motorcycles
S10_1678	23	91.87	2004-04-05	181	1:10	Motorcycles
S10_1678	59	93.79	2004-05-18	328	1:10	Motorcycles
S10_1678	34	89.00	2004-06-28	175	1:10	Motorcycles
S10_1678	45	81.35	2004-07-23	119	1:10	Motorcycles
S10_1678	36	95.70	2004-08-27	286	1:10	Motorcycles
S10_1678	23	76.56	2004-09-30	186	1:10	Motorcycles

productCode	QuantityOrdered	priceEach	orderedDate	customerNumber	product_scale	product_line
S10_1678	41	94.74	2004-10-15	121	1:10	Motorcycles
S10_1678	46	84.22	2004-11-02	157	1:10	Motorcycles
S10_1678	42	80.39	2004-11-15	131	1:10	Motorcycles
S10_1678	41	84.22	2004-11-24	382	1:10	Motorcycles
S10_1678	42	84.22	2004-12-04	323	1:10	Motorcycles
S10_1678	20	92.83	2004-12-17	282	1:10	Motorcycles
S10_1678	21	76.56	2005-02-03	119	1:10	Motorcycles
S10_1678	42	80.39	2005-03-03	462	1:10	Motorcycles
S10_1678	40	77.52	2005-04-01	496	1:10	Motorcycles
S10_1678	24	85.17	2005-04-08	201	1:10	Motorcycles
S10_1678	66	79.43	2005-05-13	141	1:10	Motorcycles
S10_1949	26	214.30	2003-01-29	121	1:10	Classic Cars
S10_1949	29	197.16	2003-03-24	144	1:10	Classic Cars

Slice

Explanation:It selects a single dimension from the OLAP cube which results in a new sub-cube creation.

In [19]:

```
%%sql
select product_vendor, month, country, count(*) as no_of_sales
from de2418_S22_classic_models_star.sales_view
group by product_vendor, month, country
having country = 'Germany'
```

```
 * mysql+pymysql://root:***@localhost
36 rows affected.
```

Out[19]:

product_vendor	month	country	no_of_sales
Autoart Studio Design	10	Germany	2
Autoart Studio Design	11	Germany	1
Welly Diecast Productions	3	Germany	1
Welly Diecast Productions	10	Germany	6
Welly Diecast Productions	11	Germany	2
Studio M Art Models	10	Germany	3

product_vendor	month	country	no_of_sales
Studio M Art Models	11	Germany	2
Highway 66 Mini Classics	10	Germany	1
Red Start Diecast	9	Germany	1
Gearbox Collectibles	10	Germany	5
Min Lin Diecast	10	Germany	3
Gearbox Collectibles	3	Germany	2
Autoart Studio Design	1	Germany	1
Gearbox Collectibles	1	Germany	1
Min Lin Diecast	9	Germany	2
Unimax Art Galleries	10	Germany	1
Min Lin Diecast	11	Germany	1
Exoto Designs	11	Germany	1
Gearbox Collectibles	11	Germany	1
Exoto Designs	9	Germany	1
Min Lin Diecast	3	Germany	1
Motor City Art Classics	11	Germany	1
Classic Metal Creations	11	Germany	2
Classic Metal Creations	10	Germany	3
Highway 66 Mini Classics	3	Germany	2
Motor City Art Classics	1	Germany	1
Classic Metal Creations	1	Germany	1
Autoart Studio Design	9	Germany	2
Second Gear Diecast	10	Germany	1
Carousel DieCast Legends	9	Germany	1
Second Gear Diecast	9	Germany	2
Exoto Designs	3	Germany	1
Studio M Art Models	9	Germany	1
Classic Metal Creations	3	Germany	1
Unimax Art Galleries	9	Germany	1
Motor City Art Classics	9	Germany	3

Dice

Explanation:It selects a sub-cube from the OLAP cube by selecting two or more dimensions.

In [20]:

```
%%sql
select product_vendor, month, country, count(*) as no_of_sales
from de2418_S22_classic_models_star.sales_view
group by product_vendor, month, country
having country in ('USA','Germany') and month in (1, 2, 3)
```

```
 * mysql+pymysql://root:***@localhost
49 rows affected.
```

Out[20]:

product_vendor	month	country	no_of_sales
Min Lin Diecast	2	USA	6
Min Lin Diecast	3	USA	9
Classic Metal Creations	1	USA	6
Classic Metal Creations	3	USA	5
Classic Metal Creations	2	USA	3
Highway 66 Mini Classics	2	USA	2
Highway 66 Mini Classics	3	USA	6
Red Start Diecast	2	USA	6
Red Start Diecast	1	USA	5
Red Start Diecast	3	USA	7
Motor City Art Classics	1	USA	9
Motor City Art Classics	2	USA	6
Second Gear Diecast	3	USA	7
Second Gear Diecast	2	USA	6
Second Gear Diecast	1	USA	5
Welly Diecast Productions	3	USA	3
Welly Diecast Productions	2	USA	5
Unimax Art Galleries	2	USA	4
Unimax Art Galleries	1	USA	3
Unimax Art Galleries	3	USA	5
Welly Diecast Productions	3	Germany	1
Exoto Designs	2	USA	5
Exoto Designs	3	USA	3
Exoto Designs	1	USA	9
Studio M Art Models	3	USA	7
Studio M Art Models	1	USA	6
Studio M Art Models	2	USA	4
Highway 66 Mini Classics	1	USA	3
Motor City Art Classics	3	USA	3

product_vendor	month	country	no_of_sales
Welly Diecast Productions	1	USA	3
Gearbox Collectibles	2	USA	4
Gearbox Collectibles	3	Germany	2
Gearbox Collectibles	1	USA	5
Autoart Studio Design	1	Germany	1
Autoart Studio Design	1	USA	6
Autoart Studio Design	3	USA	5
Carousel DieCast Legends	3	USA	7
Carousel DieCast Legends	2	USA	9
Gearbox Collectibles	1	Germany	1
Gearbox Collectibles	3	USA	4
Carousel DieCast Legends	1	USA	6
Min Lin Diecast	1	USA	3
Min Lin Diecast	3	Germany	1
Highway 66 Mini Classics	3	Germany	2
Autoart Studio Design	2	USA	6
Motor City Art Classics	1	Germany	1
Classic Metal Creations	1	Germany	1
Exoto Designs	3	Germany	1
Classic Metal Creations	3	Germany	1

Roll Up

Explanation:It is just opposite of the drill-down operation. It performs aggregation on the OLAP cube

In [21]:

```
%%sql
select product_vendor, year, region, count(*) as no_of_sales
from de2418_S22_classic_models_star.sales_view
group by product_vendor, year, region LIMIT 30;
```

```
 * mysql+pymysql://root:***@localhost
30 rows affected.
```

Out[21]:

product_vendor	year	region	no_of_sales
Min Lin Diecast	2003	None	33
Min Lin Diecast	2003	EMEA	29
Min Lin Diecast	2004	EMEA	46
Min Lin Diecast	2004	APAC	19

product_vendor	year	region	no_of_sales
Min Lin Diecast	2004	None	41
Min Lin Diecast	2005	EMEA	18
Min Lin Diecast	2005	None	13
Min Lin Diecast	2005	APAC	7
Classic Metal Creations	2003	EMEA	35
Classic Metal Creations	2003	None	39
Classic Metal Creations	2003	APAC	22
Classic Metal Creations	2004	None	45
Classic Metal Creations	2004	APAC	16
Classic Metal Creations	2004	EMEA	66
Classic Metal Creations	2005	None	15
Classic Metal Creations	2005	APAC	12
Classic Metal Creations	2005	EMEA	20
Highway 66 Mini Classics	2003	None	32
Highway 66 Mini Classics	2003	APAC	16
Highway 66 Mini Classics	2003	EMEA	27
Highway 66 Mini Classics	2004	APAC	20
Highway 66 Mini Classics	2004	None	44
Highway 66 Mini Classics	2004	EMEA	42
Highway 66 Mini Classics	2005	APAC	5
Highway 66 Mini Classics	2005	None	10
Highway 66 Mini Classics	2005	EMEA	26
Red Start Diecast	2003	None	31
Red Start Diecast	2003	APAC	12
Red Start Diecast	2003	EMEA	24
Red Start Diecast	2004	APAC	17

Drilldown

Explanation: In drill-down operation, the less detailed data is converted into highly detailed data.

In [22]:

```
%%sql
select product_vendor, year, month, region, country, count(*) as no_of_sales
from de2418_S22_classic_models_star.sales_view
group by product_vendor, year, month, region, country LIMIT 50;
```

```
 * mysql+pymysql://root:***@localhost
```

```
50 rows affected.
```

Out[22]:

product_vendor	year	month	region	country	no_of_sales
Min Lin Diecast	2003	2	None	USA	1
Min Lin Diecast	2003	5	EMEA	France	3
Min Lin Diecast	2003	7	EMEA	France	2
Min Lin Diecast	2003	8	None	USA	5
Min Lin Diecast	2003	10	None	USA	6
Min Lin Diecast	2003	11	EMEA	France	2
Min Lin Diecast	2003	11	None	None	3
Min Lin Diecast	2003	12	None	USA	4
Min Lin Diecast	2004	1	EMEA	France	3
Min Lin Diecast	2004	2	APAC	Australia	2
Min Lin Diecast	2004	4	None	USA	1
Min Lin Diecast	2004	5	None	USA	4
Min Lin Diecast	2004	6	None	USA	2
Min Lin Diecast	2004	7	EMEA	France	2
Min Lin Diecast	2004	8	None	USA	4
Min Lin Diecast	2004	9	EMEA	Finland	1
Min Lin Diecast	2004	10	EMEA	Norway	1
Min Lin Diecast	2004	11	None	USA	9
Min Lin Diecast	2004	11	APAC	Austria	1
Min Lin Diecast	2004	12	APAC	New Zealand	4
Min Lin Diecast	2004	12	APAC	Australia	2
Min Lin Diecast	2005	2	EMEA	France	2
Min Lin Diecast	2005	3	None	USA	5
Min Lin Diecast	2005	4	APAC	New Zealand	3
Min Lin Diecast	2005	4	EMEA	UK	1
Min Lin Diecast	2005	5	EMEA	Spain	1
Classic Metal Creations	2003	1	EMEA	Norway	1
Classic Metal Creations	2003	3	EMEA	Sweden	1
Classic Metal Creations	2003	5	EMEA	Spain	1
Classic Metal Creations	2003	7	None	USA	4
Classic Metal Creations	2003	9	APAC	Singapore	1
Classic Metal Creations	2003	10	None	USA	6
Classic Metal Creations	2003	11	APAC	Australia	4

product_vendor	year	month	region	country	no_of_sales
Classic Metal Creations	2003	11	None	USA	11
Classic Metal Creations	2003	11	EMEA	France	3
Classic Metal Creations	2003	12	None	Canada	1
Classic Metal Creations	2004	1	None	USA	1
Classic Metal Creations	2004	3	None	USA	2
Classic Metal Creations	2004	5	None	USA	2
Classic Metal Creations	2004	6	APAC	Japan	1
Classic Metal Creations	2004	7	APAC	Australia	1
Classic Metal Creations	2004	8	EMEA	Italy	2
Classic Metal Creations	2004	9	EMEA	Sweden	1
Classic Metal Creations	2004	10	EMEA	France	2
Classic Metal Creations	2004	10	None	USA	4
Classic Metal Creations	2004	11	None	USA	14
Classic Metal Creations	2004	11	APAC	Australia	3
Classic Metal Creations	2004	12	None	USA	3
Classic Metal Creations	2005	1	None	USA	5
Classic Metal Creations	2005	2	None	USA	1