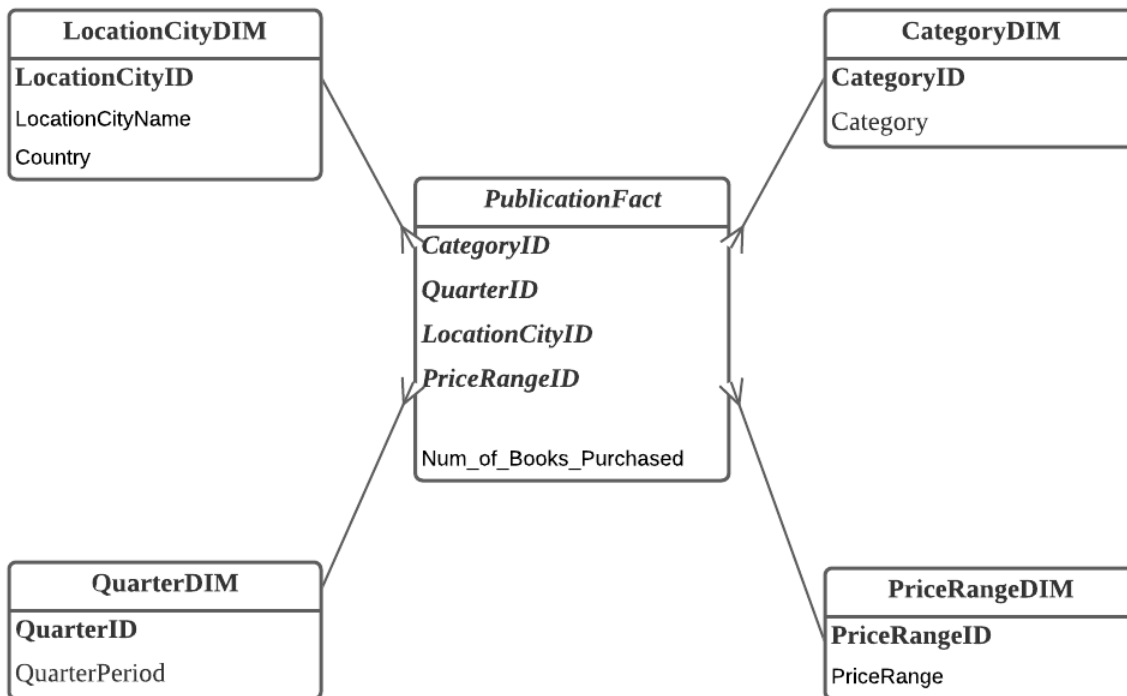


Task1:



Task 2:

| LocationCity | Total Sales |
|--------------|-------------|
| Melbourne | 156 |
| Sydney | 126 |

| Quarter | Total Sales |
|------------|-------------|
| Jan - Mar | 131 |
| Apr - Jun | 137 |
| July - Sep | 14 |
| Oct - Dec | 0 |

| Category | TotalSales |
|-----------------|------------|
| Fiction | 47 |
| Fantasy | 61 |
| Action | 55 |
| Science Fiction | 59 |
| Novels | 60 |

| PriceRange | TotalSales |
|-----------------------|------------|
| price < \$20 | 81 |
| \$20 <= price <= \$50 | 114 |
| price > \$50 | 87 |

Task 3:

table QuarterDIM Code:

```
--table QuarterDIM
```

```
create table QuarterDIM(
```

```
    QuarterID number(1),
```

```
    QuarterPeriod varchar2(20));
```

```
--insert values manually
```

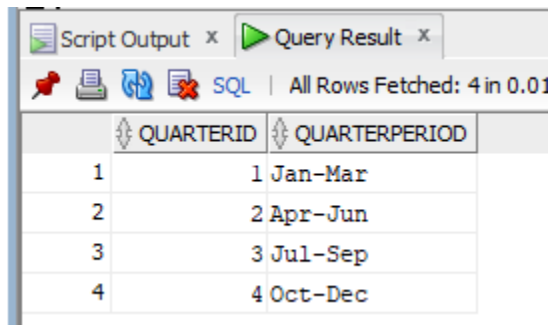
```
insert into QuarterDIM values (1, 'Jan-Mar');
```

```
insert into QuarterDIM values (2, 'Apr-Jun');
```

```
insert into QuarterDIM values (3, 'Jul-Sep');
```

```
insert into QuarterDIM values (4, 'Oct-Dec');
```

table QuarterDIM Screenshots:



The screenshot shows a SQL query result window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying a table with two columns: 'QUARTERID' and 'QUARTERPERIOD'. The table contains four rows of data.

| QUARTERID | QUARTERPERIOD |
|-----------|---------------|
| 1 | 1 Jan-Mar |
| 2 | 2 Apr-Jun |
| 3 | 3 Jul-Sep |
| 4 | 4 Oct-Dec |

table PriceRangeDIM Code:

```
--table PriceRangeDIM
```

```
create table PriceRangeDIM(
```

```
    PriceRangeID varchar2(1),
```

```
    PriceRange varchar2(20));
```

```
--insert values manually
```

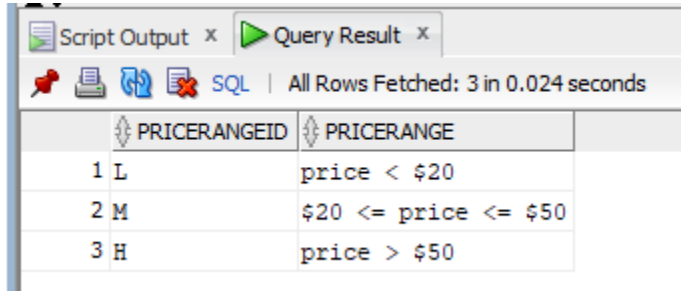
--L represents Low, M represents Medium, H represents High

insert into PriceRangeDIM values ('L', 'price < \$20');

insert into PriceRangeDIM values ('M', '\$20 <= price <= \$50');

insert into PriceRangeDIM values ('H', 'price > \$50');

table PriceRangeDIM Screenshots:



The screenshot shows a database query result window with two tabs: 'Script Output' and 'Query Result'. The 'Query Result' tab is active, displaying a table with two columns: 'PRICERANGEID' and 'PRICERANGE'. The table contains three rows of data. Above the table, it says 'All Rows Fetched: 3 in 0.024 seconds'.

| PRICERANGEID | PRICERANGE |
|--------------|-----------------------|
| 1 L | price < \$20 |
| 2 M | \$20 <= price <= \$50 |
| 3 H | price > \$50 |

table CategoryDIM Code:

--table CategoryDIM

create table CategoryDIM as

select distinct BookCategory as Category from PUBLISH.BOOK1;

--add a surrogate key

alter table CategoryDIM add (CategoryID number(3));

--use sequence to add surrogate key

create sequence category_seq_id

start with 1

increment by 1

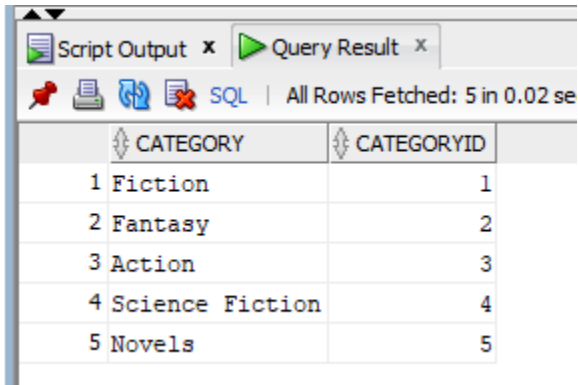
maxvalue 99999999

minvalue 1

nocycle;

update CategoryDIM set CategoryID = category_seq_id.nextval;

table CategoryDIM Screenshots:



| | CATEGORY | CATEGORYID |
|---|-----------------|------------|
| 1 | Fiction | 1 |
| 2 | Fantasy | 2 |
| 3 | Action | 3 |
| 4 | Science Fiction | 4 |
| 5 | Novels | 5 |

table LocationCityDIM Code:

```
--table LocationCityDIM
```

```
create table LocationCityDIM as
```

```
    select distinct CustomerCity as LocationCityName, CustomerCountry as Country
    from PUBLISH.CUSTOMER1;
```

```
--add a surrogate key
```

```
alter table LocationCityDIM add (LocationCityID number(3));
```

```
--use sequence to add surrogate key
```

```
create sequence location_seq_id
```

```
    start with 1
```

```
    increment by 1
```

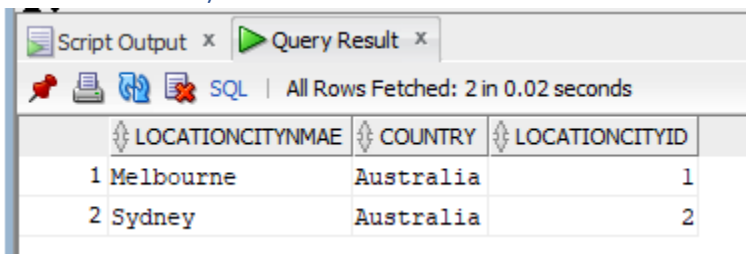
```
    maxvalue 99999999
```

```
    minvalue 1
```

```
    nocycle;
```

```
update LocationCityDIM set LocationCityID = location_seq_id.nextval;
```

table LocationCityDIM Screenshots:



| | LOCATIONCITYNAME | COUNTRY | LOCATIONCITYID |
|---|------------------|-----------|----------------|
| 1 | Melbourne | Australia | 1 |
| 2 | Sydney | Australia | 2 |

table PublicationFact Code:

```
--table PublicationFact

--use TempFact Table because we have manually created attribute

create table TempFact as

select t.transactiondate, b.price, b.bookcategory, c.customercity, t.quantity

from PUBLISH.book1 b, PUBLISH.transaction1 t, PUBLISH.customer1 c

where t.bookisbn = b.bookisbn and t.customerid = c.customerid;

--add QuarterID to TempFact table

alter table TempFact add(QuarterID number(1));

--update QuarterID value in TempFact table

update TempFact set QuarterID = 1

where to_char(transactiondate, 'MM') >= '01'

and to_char(transactiondate, 'MM') <= '03';

update TempFact set QuarterID = 2

where to_char(transactiondate, 'MM') >= '04'

and to_char(transactiondate, 'MM') <= '06';

update TempFact set QuarterID = 3

where to_char(transactiondate, 'MM') >= '07'

and to_char(transactiondate, 'MM') <= '09';

update TempFact set QuarterID = 4

where QuarterID is null;

--add PriceRangeID to TempFact table

alter table TempFact add (PriceRangeID varchar2(1));

--update PriceRangeID value in TempFact table

update TempFact set PriceRangeID = 'L'

where price < 20;
```

```
update TempFact set PriceRangeID = 'H'
```

```
where price > 50;
```

```
update TempFact set PriceRangeID = 'M'
```

```
where PriceRangeID is null;
```

```
--add CategoryID to TempFact table
```

```
alter table TempFact add(CategoryID number(3));
```

```
--update CategoryID value in TempFact table
```

```
update TempFact tf
```

```
set tf.CategoryID = (select c.categoryid
```

```
from CategoryDIM c
```

```
where c.category = tf.bookcategory);
```

```
--add LocationCityID to TempFact table
```

```
alter table TempFact add(LocationCityID number(3));
```

```
--update LocationCityID value in TempFact table
```

```
update TempFact tf
```

```
set tf.LocationCityID = (select l.locationcityid
```

```
from LocationCityDIM l
```

```
where l.locationcityname = tf.customercity);
```

```
--create actual PublicationFact table
```





```
create table PublicationFact as
```

```
select CategoryID, QuarterID, LocationCityID, PriceRangeID, sum(Quantity) as Num_of_Books_Purchased
```

```
from TempFact
```

```
group by CategoryID, QuarterID, LocationCityID, PriceRangeID;
```

table PublicationFact Screenshots (does not include all records due to the amount of records):

| Script Output x | | Query Result x | | | |
|---|---|---|---|--------------|------------------------|
|  |  |  |  SQL All Rows Fetched: 40 in 0.022 seconds | | |
| | CATEGORYID | QUARTERID | LOCATIONCITYID | PRICERANGEID | NUM_OF_BOOKS_PURCHASED |
| 1 | 2 | 2 | 2 | M | 6 |
| 2 | 3 | 2 | 2 | M | 7 |
| 3 | 3 | 2 | 1 | H | 6 |
| 4 | 5 | 4 | 2 | L | 4 |
| 5 | 1 | 1 | 2 | L | 3 |
| 6 | 2 | 2 | 1 | L | 1 |
| 7 | 2 | 1 | 1 | L | 2 |
| 8 | 4 | 2 | 1 | H | 12 |
| 9 | 2 | 4 | 2 | M | 2 |
| 10 | 3 | 2 | 1 | M | 8 |
| 11 | 3 | 2 | 2 | H | 3 |
| 12 | 1 | 2 | 1 | L | 11 |
| 13 | 2 | 1 | 2 | M | 14 |
| 14 | 5 | 2 | 1 | L | 11 |
| 15 | 1 | 2 | 2 | M | 2 |
| 16 | 3 | 1 | 1 | H | 10 |
| 17 | 4 | 4 | 2 | H | 1 |
| 18 | 3 | 4 | 2 | M | 2 |

Task 4:

--a) Show total number of books sold by different transaction periods in Melbourne.

```
select p.QuarterID, q.QuarterPeriod, sum(Num_of_Books_Purchased) as Total_Num_of_Books_Sold
from PublicationFact p, QuarterDIM q, LocationCityDIM l
where p.QuarterID = q.QuarterID and p.LocationCityID = l.LocationCityID
and l.LocationCityName = 'Melbourne'
group by p.QuarterID, q.QuarterPeriod
order by p.QuarterID;
```

| Query Result x | | | |
|--|-----------|---------------|-------------------------|
| SQL All Rows Fetched: 3 in 0.027 seconds | | | |
| | QUARTERID | QUARTERPERIOD | TOTAL_NUM_OF_BOOKS_SOLD |
| 1 | 1 | Jan-Mar | 78 |
| 2 | 2 | Apr-Jun | 73 |
| 3 | 4 | Oct-Dec | 5 |

--b) Show total number of books sold by each book category

```
select p.CategoryID, c.Category, sum(Num_of_Books_Purchased) as Total_Num_of_Books_Sold
from PublicationFact p, CategoryDIM c
where p.CategoryID = c.CategoryID
group by p.CategoryID, c.Category
order by p.CategoryID;
```

| Query Result x | | | |
|--|----------|-----------------|-------------------------|
| SQL All Rows Fetched: 5 in 0.018 seconds | | | |
| | CATEG... | CATEGORY | TOTAL_NUM_OF_BOOKS_SOLD |
| 1 | 1 | Fiction | 47 |
| 2 | 2 | Fantasy | 61 |
| 3 | 3 | Action | 55 |
| 4 | 4 | Science Fiction | 59 |
| 5 | 5 | Novels | 60 |

--c) Show total number of Fantasy books sold below \$20

```
select sum(Num_of_Books_Purchased) as Total_Num_of_Books_Sold
from PublicationFact p, CategoryDIM c
where p.CategoryID = c.CategoryID
and c.Category = 'Fantasy' and p.PriceRangeID = 'L';
```

| Query Result x | | |
|--------------------------------------|-------------------------|--|
| SQL All Rows Fetched: 1 in 0.018 s | | |
| | TOTAL_NUM_OF_BOOKS_SOLD | |
| 1 | 3 | |

Task 5

--a)show total number of books sold by different book category in Sydney

--Reason for this query:

--the manager may want to know how different categories of books sell in a certain location

--to help further decision like put more books that from a category that sells a lot

```
select p.CategoryID, c.Category, sum(Num_of_Books_Purchased) as Total_Num_of_Books_Sold
```

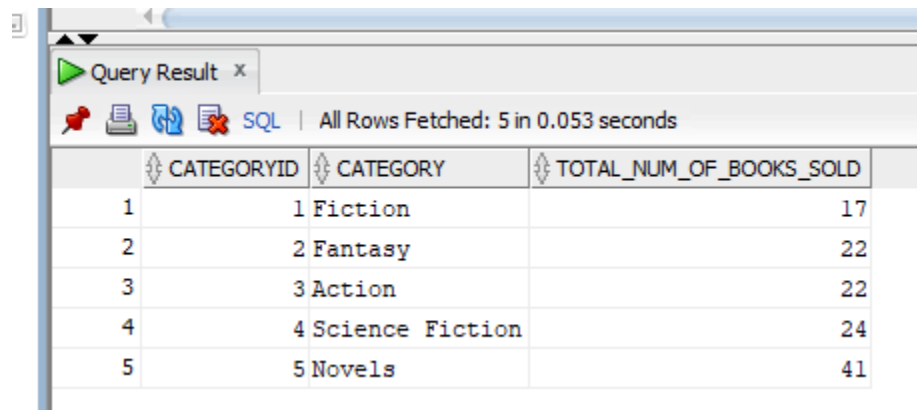
```
from PublicationFact p, CategoryDIM c, LocationCityDIM l
```

```
where p.CategoryID = c.CategoryID and p.LocationCityID = l.LocationCityID
```

```
and l.LocationCityName = 'Sydney'
```

```
group by p.CategoryID, c.Category
```

```
order by p.CategoryID;
```



The screenshot shows a SQL query result window titled "Query Result x". It displays the results of the query for books sold in Sydney, grouped by category. The window includes a toolbar with icons for saving, printing, refreshing, and executing queries, along with a status bar indicating "All Rows Fetched: 5 in 0.053 seconds".

| | CATEGORYID | CATEGORY | TOTAL_NUM_OF_BOOKS_SOLD |
|---|------------|-----------------|-------------------------|
| 1 | 1 | Fiction | 17 |
| 2 | 2 | Fantasy | 22 |
| 3 | 3 | Action | 22 |
| 4 | 4 | Science Fiction | 24 |
| 5 | 5 | Novels | 41 |

--b)show total number of Novels books sold in Melbourne from different quarter

--Reason for this query:

--the manager may want to compare how a certain category of book sold in a certain location in different period

--to help further decision like having discount for that category of books when the total sales are low

```
select q.QuarterID, q.QuarterPeriod, sum(Num_of_Books_Purchased) as Total_Num_of_Books_Sold
```

```
from PublicationFact p, CategoryDIM c, LocationCityDIM l, QuarterDIM q
```

```
where p.CategoryID = c.CategoryID
```

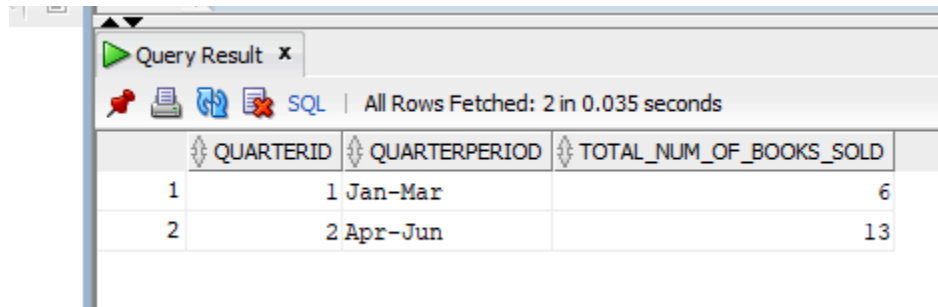
```
and p.LocationCityID = l.LocationCityID
```

and p.QuarterID = q.QuarterID

and c.Category = 'Novels' and l.LocationCityName = 'Melbourne'

group by q.QuarterID, q.QuarterPeriod

order by q.QuarterID;



Query Result x

SQL | All Rows Fetched: 2 in 0.035 seconds

| QUARTERID | QUARTERPERIOD | TOTAL_NUM_OF_BOOKS_SOLD |
|-----------|---------------|-------------------------|
| 1 | 1 Jan-Mar | 6 |
| 2 | 2 Apr-Jun | 13 |