Database Management Systems - UE19CS301 Assignment-3

Date: 31/10/2021

Section: H

Sumukh Raju Bhat	PES1UG19CS519
Shreyas VBKS	PES1UG19CS469
Sowmya M	PES1UG19CS497

Users and privileges:

Assume that the 3 users are numbered for simplicity of explanations:

po_shipment_mgr: 1 po_products_mgr: 2 carrier_shipment_mgr: 3

For each table in the database, the following permissions are given to po_shipment_mgr, po_products_mgr, carrier_shipment_mgr mentioned as serial number 1, 2 and 3 respectively:

requests:

- 1. insert, update, select, delete
- 2. -
- 3. select

appointment:

- 1. select, update(status)
- 2. -
- 3. insert, update, select, delete

orders:

- 1. insert, update, select, delete
- 2. -
- 3. select

items, products:

1. select

- 2. select, update, insert, delete
- 3. -

carrier_c_contact:

- 1. select
- 2. -
- 3. select, insert, update, delete

purchasing_organization_po_contact:

- 1. select, insert, update, delete
- 2. select, insert, update, delete
- 3. select

location:

- 1. select, insert, update, delete
- 2. select, insert, update, delete
- 3. select

location I contact:

- 1. select, insert, update, delete
- 2. select, insert, update, delete
- 3. select

carrier.

- 1. select
- 2. select
- 3. insert, update, select, delete

purchasing_organization:

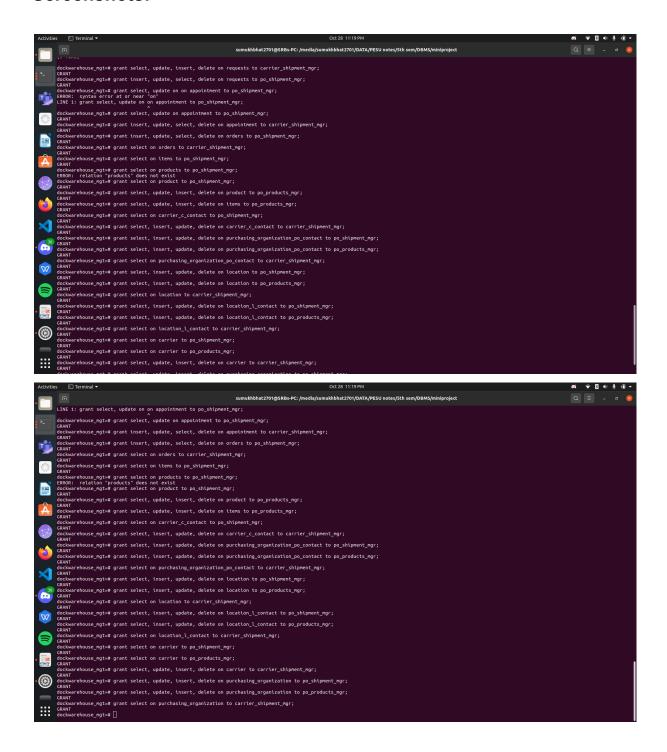
- 1. insert, update, select, delete
- 2. insert, update, select, delete
- 3. select

```
Code:
```

```
create user po shipment mgr with password 'po@123';
create user po products mgr with password 'po@789';
create user carrier shipment mgr with password 'car@456';
grant select, update, insert, delete on requests to carrier shipment mgr;
grant insert, update, select, delete on requests to posshipment mgr;
grant select, update on appointment to po shipment mgr;
grant insert, update, select, delete on appointment to
carrier shipment mgr;
grant insert, update, select, delete on orders to po shipment mgr;
grant select on orders to carrier shipment mgr;
grant select on items to po shipment mgr;
grant select on product to po shipment mgr;
grant select, update, insert, delete on product to po products mgr;
grant select, update, insert, delete on items to po_products_mgr;
grant select on carrier c contact to po shipment mgr;
grant select, insert, update, delete on carrier c contact to
carrier shipment mgr;
grant select, insert, update, delete on
purchasing organization po contact to po shipment mgr;
```

```
grant select, insert, update, delete on
purchasing organization po contact to po products mgr;
grant select on purchasing organization po contact to
carrier_shipment_mgr;
grant select, insert, update, delete on location to po shipment mgr;
grant select, insert, update, delete on location to po products mgr;
grant select on location to carrier_shipment_mgr;
grant select, insert, update, delete on location 1 contact to
po_shipment_mgr;
grant select, insert, update, delete on location 1 contact to
po_products_mgr;
grant select on location I contact to carrier shipment mgr;
grant select on carrier to po_shipment_mgr;
grant select on carrier to po products mgr;
grant select, update, insert, delete on carrier to carrier shipment mgr;
grant select, update, insert, delete on purchasing_organization to
po_shipment_mgr;
grant select, update, insert, delete on purchasing organization to
po_products_mgr;
grant select on purchasing organization to carrier shipment mgr;
```

Screenshots:



VERIFICATION OF SOME OF THE PERMISSIONS:

User 1: po_shipment_mgr

```
Username [postgres]: po_shipment_mgr
Password for user po_shipment_mgr:
psql (13.4)
WARNING: Console code page (437) differs from Windows code page (1252)
        8-bit characters might not work correctly. See psql reference
        page "Notes for Windows users" for details.
Type "help" for help.
postgres=> \c dockwarehouse_mgt;
You are now connected to database "dockwarehouse_mgt" as user "po_shipment_mgr".
dockwarehouse mgt=> select * from requests;
 deadline | po_id | carrier_id
2020-10-02
                             22
2020-11-06
                 2
                             33
2020-12-04
                 9
                             44
2021-10-07
                 8
                             55
                 4
                             99
2020-10-05
                             25
2020-10-10
                 2
                              26
                             66
(8 rows)
```

order_id	alloted_date	dock	total_quantity	special_instrs	po_id	carrier_id	location_id
801	2020-10-01	1	600		1	22	111
802	2020-10-01	2	800		1	33	121
803	2020-10-02	5	1100		2	44	125
804	2020-10-02	1	225	Glass packing, Careful!	2	55	125
805	2020-10-03	1	350		3	99	127
806	2020-10-04	1	400		4	22	128
807	2020-10-05	1	500		5	33	111
808	2020-10-06	1	600		6	44	121
(8 rows)							

```
dockwarehouse_mgt=> delete from carrier where carrier_id=22;
ERROR: permission denied for table carrier
```

dockwarehouse_mgt=> delete from carrier_c_contact where carrier_id=33; ERROR: permission denied for table carrier_c_contact

User 2: po_products_mgr

	e_mgt=> select ^a product_name	
451	Chocolates	1
452	Noodles	j 1
455	Laptops	2
456	Television	2
457	Cars] 3
458	Snacks	j 4
459	Cold Drinks	5
450	Lotions	6
453	Serials	j 7
461	Cholcolates	8
462	Cold Drinks	9
(11 rows)		

dockwareho	use_mgt=> select	* from item	ns;		
item_id	dangerous_good	quantity 	weight +	order_id	product_id
211	0	500	60.12	801	451
212	0	600	21.32	802	451
213	1	1000	32.12	803	452
214	1	200	45.55	804	452
215	0	300	100	805	452
216	1	400	120.56	806	452
217	0	500	62.21	807	455
218	1	600	63.03	808	455
219	1	100	51.56	801	456
210	1	2000	53.89	802	457
209	0	100	56.56		458
208	1	25	59.78		458
207	1	50	32.91		461
(13 rows)					

```
dockwarehouse_mgt=> select * from orders;
ERROR: permission denied for table orders
```

dockwarehouse_mgt=> delete from carrier where carrier_id=33; ERROR: permission denied for table carrier

User 3: carrier_shipment_mgr

dockwarehouse	e_mgt=> select * fr	rom carrier;	0	_ '
carrier_id	c_name	c_street	c_city	c_state
	 		+	+
22	FedEx	GM Road	Bangalore	Karnataka
33	XPO	Tilak Road	Bangalore	Karnataka
44	YRC	Fort Road	Kozhikode	Kerala
55	R+L	Baga Road	Cuncolim	Goa
99	Holland	Manjappa Road	Bangalore	Karnataka
25	Read Away	Sanki Road	Bangalore	Karnataka
66	Aveerit Express	St.Marks Road	Bangalore	Karnataka
68	Forward Air	DVG Road	Bangalore	Karnataka
77	Oak Harbour	Race Course Road	Bangalore	Karnataka
26	UPS Freight	LK Road	Bangalore	Karnataka
(10 rows)				

```
dockwarehouse_mgt=> select * from carrier_c_contact;
c_contact | carrier_id
9112345678
                     22
8278654321
                     33
9312137891
                     44
                     55
9434148921
8556157819
                     99
8678167911
                     25
7791177777
                     26
9801189991
                     66
9911192832
                     68
9101220910
                     77
911314159
                     26
8121314156
                     77
(12 rows)
```

```
dockwarehouse_mgt=> select * from items;
ERROR: permission denied for table items
```

dockwarehouse_mgt=> delete from purchasing_organization_po_contact where po_id=4; ERROR: permission denied for table purchasing_organization_po_contact

Simple Queries:

1. Find out how many orders are there for a particular PO on a given day.

```
select count(*) from orders where po_id = 1 and orders.alloted_date = '2020-10-01';
```

```
count
-----
2
(1 row)
```

2. Find out which docks are occupied for a particular PO on a given day.

```
select count(*) from orders where po_id = 1 and orders.alloted_date = '2020-10-01' group by po_id;
```

```
count
-----
2
(1 row)
```

3. Find out the appointments of a particular carrier which are approved and by a PO.

select po_id, appointment_id from appointment where carrier_id = 33 and status = 1;

4. Find out which appointments of a particular carrier are rejected and by which PO.

select po_id, appointment_id from appointment where carrier_id = 33 and status = 0;

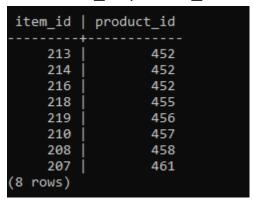
```
po_id | appointment_id
-----+
(0 rows)
```

Find out the count of items manufactured in each product.
 select product id, count(*) from items group by product id;

Find out the manufacturing locations of a particular PO.
 select I_street, I_city, I_state from location where po_id = 2;

7. List all the dangerous goods manufactured.

select item_id, product_id from items where dangerous_good = 1;



8. Get the unit weight of a particular item.

select quantity/weight as unit_weight from items where item_id =
211;

```
unit_weight
-----
8.316699933466401
(1 row)
```

Complex queries:

1. Find all the carrier names who have minimum number of orders when compared to all others

select c_name from carrier natural join (select carrier_id from (select carrier_id, count(*) from orders group by carrier_id) as y where count = (select min(count) from (select carrier_id, count(*) from orders group by carrier_id) as x)) as z;

```
c_name
-----
Holland
R+L
(2 rows)
```

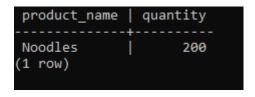
Find the total quantity of all items across all Nestle's manufacturing units

select sum(quantity) as Total_Quantity from product natural join items natural join purchasing_organization where po_name = 'Nestle';

```
total_quantity
-----
3000
(1 row)
```

3. Select product name and quantity of a product with least number of items manufactured in a particular PO.

select product_name, quantity from product natural join items where quantity = (select min(quantity) from items natural join product where po_id = 1);



4. For a particular carrier, select all those orders whose destination location is Bangalore and display to which PO such order belonged to and the stop point of the order.

select order_id, po_name, po_street from purchasing_organization as p, orders as o, location as l, carrier as c where p.po_id = o.po_id and o.location_id = l.location_id and c.carrier_id = o.carrier_id and type = 's' and l_city = 'Bangalore';

```
order_id
            po name
                           po_street
     801
           Nestle
                       Bluebonnet Cane
     802
           Nestle
                       Bluebonnet Cane
           Pepsi Co.
     807
                       Fontane Lane
     808
           Unilever
                      | Lavelle Road
 rows)
```

5. The city/cities with maximum number of orders being delivered.

select l_city from location where location_id in (select location_id from (select l.location_id, count(*) from location as l, orders as o where l.location_id = o.location_id group by l.location_id) as y where count = (select max(count) from (select l.location_id, count(*) from location as l, orders as o where l.location_id = o.location_id group by l.location_id) as x));

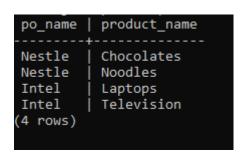
```
l_city
-----
Bangalore
Warangal
Bangalore
(3 rows)
```

6. For a particular carrier, calculate the number of dangerous items it has to deliver and which order it belongs to.

select order_id, count(*) from carrier natural join orders natural join items where dangerous_good = 1 and carrier_id = 22 group by order_id;

7. Find out which product is the most popular in terms of number of orders and specify which PO manufactures it

select po_name, product_name from (select product_name, po_id from (select product_id from (select product_id, count(*) from orders natural join product group by product_id) as x where x.count = (select max(count) from (select product_id, count(*) from orders natural join product group by product_id) as y)) as z natural join product) as t natural join purchasing organization;



Concurrency control:

Read committed isolation level:

Transaction 1:

```
begin;
set session transaction isolation level read committed;
show transaction_isolation;
select * from items;
update items set quantity = 450 where item_id = 216;
commit;
```

Transaction 2:

```
begin;
set session transaction isolation level read committed;
show transaction_isolation;
select * from items where item_id=216;
select * from items where item_id=216;
```

- In read committed level the changes made to a table are not reflected unless the transaction is committed.
- Here in transaction 1 we change the quantity field of the item table for item id=216.
- Before committing the change we check the value of quantity of that particular item id in transaction 2 and the output displayed is 400 which is the old value.
- Now in transaction 1 we commit the change by the command commit.
- Therefore the second time we check for the quantity field of item_id 216 in transaction 2 we get the updated field which is 450.

Transaction 1:

```
C:\WINDOWS\system32\cmd.exe
dockwarehouse_mgt=# begin;
BEGIN
dockwarehouse_mgt=*# set session transaction isolation level read committed;
SET
dockwarehouse_mgt=*# show transaction_isolation;
transaction_isolation
read committed
(1 row)
dockwarehouse_mgt=*# select * from items;
item_id | dangerous_good | quantity | weight | order_id | product_id
                                      500
                                              60.12
21.32
                                                              801
                                                                             451
                           0
                                      600
                                                              802
     213
                                     1000
                                              32.12
                                                              803
                                              45.55
                                                                             452
     214
                                      200
                                                              804
                                      300
                                                 100
                                                              805
     216
                                      400
                                              120.56
                                                              806
                                                                             452
                                              62.21
63.03
                           0
                                                                             455
                                      500
     217
                                                              807
     218
                                      600
                                                              808
     219
                                      100
                                              51.56
                                                              801
                                                                             456
                                              53.89
                                     2000
                                                              802
                                                                             457
     210
     209
                                      100
                                               56.56
                                                                             458
     208
                                       25
                                               59.78
                                                                             458
                                       50
                                                                             461
                                              32.91
     207
(13 rows)
dockwarehouse_mgt=*# update items set quantity = 450 where item_id = 216;
dockwarehouse_mgt=*# select * from items where item_id=216;
item_id | dangerous_good | quantity | weight | order_id | product_id
     216 |
                                      450 | 120.56 |
                                                              806 |
                                                                             452
1 row)
dockwarehouse_mgt=*# commit;
```

Transaction 2:

```
C:\WINDOWS\system32\cmd.exe
ROLLBACK
dockwarehouse mgt=# begin;
BEGIN
dockwarehouse_mgt=*# set session transaction isolation level read committed;
SET
dockwarehouse_mgt=*# show transaction_isolation;
transaction isolation
read committed
(1 row)
dockwarehouse_mgt=*# select * from items where item_id=216;
item_id | dangerous_good | quantity | weight | order_id | product_id
                         1 |
                                  400 | 120.56 |
     216
                                                        806
                                                                      452
(1 row)
dockwarehouse_mgt=*# select * from items where item_id=216;
item_id | dangerous_good | quantity | weight | order_id | product_id
                                 400 | 120.56 |
     216
                                                       806
                                                                      452
(1 row)
dockwarehouse_mgt=*# commit;
dockwarehouse_mgt=# select * from items where item_id=216;
item_id | dangerous_good | quantity | weight | order_id | product_id
     216
                         1 |
                                   450 | 120.56 |
                                                        806 l
                                                                      452
1 row)
```

Serialization isolation level:

Transaction 1:

```
begin;
show transaction isolation level;
set transaction isolation level serialization;
show transaction level serialization;
Select * from location;
insert into location(I_street, I_city, I_state, type, location_id) values
('ABK','Mumbai','Maharashtra','s',136);
select * from location;
commit;
```

Transaction 2:

```
begin;
show transaction isolation level;
set transaction isolation level serialization;
show transaction level serialization;
Select * from location;
select * from location;
insert into location(I_street, I_city, I_state, type, location_id) values
('ABK','Mumbai','Maharashtra','s',136);
```

- Serialization isolation level executes the queries of different transactions as if it is executed serially one by one rather than concurrently.
- Here in transaction 1 after setting the isolation level to serialization in both the transaction, insertion to the table location is done.
- After the successful execution of this query the table locations have been updated in transaction 1 but not in transaction 2. Now if we try to insert into transaction 1, the command doesn't execute until the transaction 1 is committed.
- But the execution gives us an error that this command has read/write dependencies among transactions and if retried it would work.
- Hence serialisation makes it look like the commands are being executed in serial order and monitors over the error that can occur due to conflicts.

Note: On the left is Transaction 1 and on the right is Transaction 2

dockwarehouse_m BEGIN dockwarehouse_m SET dockwarehouse m	gt=# set trans			el serializab ¹	le;	dockwarehouse_m BEGIN dockwarehouse_m SET dockwarehouse_m	gt=# set tran: gt=# select *	from location				
l_street	l_city			location_id	po_id	l_street	l_city +			location_id	po_id	
Malleshwaram EC Kormangala Whitefeild Bayappanalii Afzad khulm Sultan Bazar Kaveri Street MKB Road JFK Road SQL Road Avenue Street MG Road SP Road Levelle Road	Pune Pune Pune Coimbature	Karnataka Karnataka Karnataka Karnataka Telangana Telangana Tamil Nadu Maharastra Maharastra Tamil Nadu Karnataka Karnataka	s p p	111 121 122 123 124 125 126 127 128 129 130 131 132 133 134	1 2 3 4 5 6 7 8 9	Malleshwaram EC Kormangala Whitefeild Bayappanalli Afzad khulm Sular Bazar Kaveri Street MKB Road JFK Road SQL Road SQL Road Avenue Street MKB Road JESU Road Avenue Street MKB Road Levelle Road (16 rows)	Bangalore Bangalore Bangalore Bangalore Bangalore Bangalore Warangal Warangal Chennai Pune Pune Pune Bangalore Bangalore Bangalore Bangalore Bangalore	Karnataka Karnataka Karnataka Karnataka Karnataka Telangana Tamil Nadu Maharastra Maharastra Tamil Nadu Karnataka Karnataka Karnataka Karnataka	s s p p s s s	111 121 122 123 124 125 126 127 128 129 130 131 132 133 134	1 1 2 3 3 4 4 5 5 6 1 7 7 8 8 9 2 2	
values ('ABK ['] , INSERT 0 1 dockwarehouse_m	"Mumbai",'Maha gt=# select *	arashtra','s', from location	, 1 36); n;	, l_city, l_si location_id 111 121 122 123 124		dockwarehouse_m l_street Malleshwaram EC Kormangala Whitefeild Bayappanalli Afzad khulm Sultan Bazar Kaveri Street MKB Road			type s s p p p s s	location_id 111 121 122 123 124 125 126 127	po_id + 1 2 3	

SQG Road	l Pune	Maharastra	р	130	5		MG Road	Bangalore	Karnataka	l p	133	8	
SQL Road	Coimbature	Tamil Nadu	p i	131			SP Road	Bangalore	Karnataka	j p	134		
Avenue Street	Bangalore	Karnataka	p i	132			Levelle Road	Bangalore	Karnataka	j p	135		
MG Road	Bangalore	Karnataka	p i	133	8		(16 rows)						
SP Road	Bangalore	Karnataka	p j	134									
Levelle Road	Bangalore	Karnataka	p i	135			dockwarehouse m	gt=# select *	from locatio	n;			
(16 rows)							l_street _		l_state	type	location_id	po_id	
dockwarehouse m	at=# insert i	nto location(1	street	1 city 1 sta	ate type	location id)	Malleshwaram	+ Bangalore	+ Karnataka	+ s	+ 111		
values ('ABK',				,	ice, cype,	tocation_ia/	EC	Bangalore	Karnataka	S	121		
INSERT 0 1	Humbur , Hum	arasiicia , s ,	1307,				Kormangala	Bangalore	Karnataka	D	122		
dockwarehouse m	nt=# select *	from location					Whitefeild	Bangalore	Karnataka	l p	123	2	
l street	l city			location id	no id		Bayappanalli	Bangalore	Karnataka	p	124		
	+	+	+				Afzad khulm	Warangal	Telangana	P	125		
Malleshwaram	Bangalore	Karnataka	l s	111			Sultan Bazar	Warangal	Telangana	S	126		
EC	Bangalore	Karnataka	S	121			Kaveri Street		Tamil Nadu		127		
Kormangala	Bangalore	Karnataka	p	122			MKB Road	Pune	Maharastra		128		
Whitefeild	Bangalore	Karnataka	p	123	2		JFK Road	Pune	Maharastra		129	4	
Bayappanalli	Bangalore	Karnataka	p	124	3		SOG Road	Pune	Maharastra		130	5	
Afzad khulm	Warangal	Telangana	S	125			SOL Road	Coimbature	Tamil Nadu		131	6	
Sultan Bazar	Warangal	Telangana	İs	126			Avenue Street	Bangalore	Karnataka	j p	132		
Kaveri Street	Chennai	Tamil Nadu	S	127			MG Road	Bangalore	Karnataka	į p	133	8	
MKB Road	Pune	Maharastra	İs	128			SP Road	Bangalore	Karnataka	j p	134	9	
JFK Road	Pune	Maharastra	i p	129			Levelle Road	Bangalore	Karnataka	i p	135		
SQG Road	Pune	Maharastra	j p	130			(16 rows)						
SQL Road	Coimbature	Tamil Nadu	j p	131									
Avenue Street	Bangalore	Karnataka	j p	132			dockwarehouse m	gt=# insert i	nto location(l stree	t, l city, l s	tate, type, locat	ion id
MG Road	Bangalore	Karnataka	j p	133) values ('ABK'						
SP Road	Bangalore	Karnataka	j p	134			ERROR: could no	ot serialize	access due to	read/w	rite dependenc:	ies among transac	tions
Levelle Road	Bangalore	Karnataka	j p	135			DETAIL: Reason	code: Cancel	ed on identif	ication	as a pivot, du	uring conflict in	check
ABK	Mumbai	Maharashtra		136			ing.						
(17 rows)							HINT: The trans						
							dockwarehouse_m						
dockwarehouse_m								transaction	is aborted, c	ommands	ignored until	end of transacti	on blo
dockwarehouse_m	gt=# commit;						ck						
COMMIT							dockwarehouse_m						
dockwarehouse_m	gt=#						dockwarehouse_m	gt=#					

Team contributions:

- 1. Sumukh Bhat
 - Creating users and granting permissions(30mins), 2 complex queries(30mins), 4 simple queries(30mins)

2. Shreyas VBKS

 Demonstration of read committed isolation level(45mins), 3 complex queries(30mins), 2 simple queries(15mins)

3. Sowmya M

 Demonstration of serialization isolation level(45mins), 3 complex queries(30mins), 2 simple queries(15mins)