Lab-9 SQL Queries

Prepared by:

Nisarg Bhalia (201901220)

Sarthak Patel (201901260)

Jitanshu Shaw (201901292)

Meet Sable (201901442)

Group 9, Section 8

20 Nov Tuesday, 2021

Queries

1. select * from employee where shift_hours = 5;

Data Output | Explain | Messages | Notifications

	employee_id [PK] bigint	name character (200)	password character (12)	age bigint	emplyee_type_id bigint	joined_since date	prefered_work_hour time without time zone	shift_hours bigint	contact_id bigint
1	1	Wileen ...	3nIgxYP6dKA	42	1	2020-11-04	11:55:00	5	1
2	5	Janifer ...	U1wsjaYYUNj	46	5	2019-03-18	23:30:00	5	5
3	20	Abbey ...	XjnJiFKV8	18	20	2021-10-20	11:10:00	5	20
4	22	Sharia ...	HANiYWFqXACE	32	2	2018-12-15	14:03:00	5	22
5	26	Anastasie ...	dRdBsebGK	18	6	2021-02-12	17:28:00	5	26
6	30	Any ...	Fw1YXAN	45	10	2021-08-07	13:01:00	5	30
7	47	Dael ...	15ElwPV	33	7	2020-05-01	12:16:00	5	47
8	48	Yolane ...	efNISJvneQ	35	8	2019-07-28	20:24:00	5	48
9	49	Rikki ...	GI99g1ElRnW	36	9	2018-12-31	13:35:00	5	49
10	62	Charissa ...	Bq7xe634	49	2	2019-02-08	14:24:00	5	62
11	93	Hodge ...	68UXdf0	37	13	2020-06-21	23:11:00	5	93
12	96	Hildegarde ...	ndDTetGEg	42	16	2020-06-25	23:38:00	5	96
13	98	Carly ...	YuyMQqAIZqho	35	18	2019-11-27	16:22:00	5	98

2. select name, age from employee where extract(year from joined_since) between 2019 and 2020;

Data Output | Explain | Messages | Notifications

	name character (200)	age bigint
40	Chelsea ...	33
41	Kermie ...	34
42	Eustacia ...	22
43	Tomlin ...	39
44	Danni ...	21
45	Pegeen ...	33
46	Letty ...	25
47	Vina ...	29
48	Arlee ...	43
49	Bartel ...	49
50	Cullen ...	46
51	Porty ...	29
52	Darin ...	49
53	Jodi ...	25
54	Gib ...	21
55	Mikol ...	36
56	Cull ...	24
57	Hodge ...	37
58	Ashla ...	47
59	Hildegarde ...	42
60	Carly ...	35
61	Kaleb ...	25
62	Donny ...	27

3. SELECT name, age FROM employee where age >35 GROUP BY name, age ORDER BY name;

Data Output | Explain | Messages | Notifications

	name character (200)	age bigint
25	Hildegarde	42
26	Hodge	37
27	Ilario	38
28	Janifer	46
29	Katine	43
30	Kimberley	49
31	Loren	40
32	Malvina	36
33	Mikol	36
34	Mufi	50
35	Nicola	46
36	Rafe	43
37	Renata	38
38	Rikki	36
39	Rosabel	37
40	Rosie	39
41	Ruby	49
42	Saunderson	39
43	Tabbitha	48
44	Tersina	45
45	Tomlin	39
46	Wileen	42
47	Winn	50

4. SELECT * FROM department;

Data Output | Explain | Messages | Notifications

	department [PK] character (100)
1	Legal
2	Engineering
3	Sales
4	Marketing
5	Support
6	Research and Development
7	Human Recources
8	Accounting
9	Product
10	Management

5. select * from employee where extract(hour from prefered_work_hour) = 12;

Data Output | Explain | Messages | Notifications

	employee_id [PK] bigint	name character (200)	password character (12)	age bigint	emplyee_type_id bigint	joined_since date	prefered_work_hour time without time zone	shift_hours bigint	contact_id bigint
1	15	Georges ...	nDL8E9Jlt	32	15	2019-11-03	12:18:00	8	15
2	21	Zared ...	RMjvrNBHs	21	1	2019-11-15	12:51:00	6	21
3	27	Anthea ...	0kknKVIvjG	27	7	2021-10-17	12:25:00	2	27
4	41	Malvina ...	vTWVZd	36	1	2020-05-18	12:32:00	2	41
5	45	Rosie ...	g8SitXW	39	5	2021-03-05	12:18:00	2	45
6	47	Dael ...	15ElwPV	33	7	2020-05-01	12:16:00	5	47
7	59	Augusta ...	sujx7IKw472	43	19	2021-04-04	12:59:00	3	59
8	60	Karel ...	T0gjvEkiW	34	20	2021-02-01	12:01:00	6	60
9	66	Kermie ...	0m6m4ckW	34	6	2019-06-01	12:45:00	7	66
10	88	Gib ...	ocOtruh6oZaz	21	8	2020-04-06	12:53:00	4	88

6. Select sum(current_stock) from toy_base;

Data Output | Explain | Messages | Notifications

	sum numeric
1	2654

7. select count(employee_id) from employee where age >40;

	count bigint
1	32

8. SELECT AVG(current_stock)FROM raw_materials;

Data Output | Explain | Messages | Notifications

	count bigint
1	32

9. CREATE VIEW Old_employees AS SELECT name, employee_id FROM employee WHERE age> 40; select * from Old_employees;

Data Output | Explain | Messages | Notifications

	name character (200)	employee_id bigint
10	Hildagard	31
11	Winn	32
12	Ambros	33
13	Derrick	34
14	Tabbitha	35
15	Rafe	36
16	Katine	37
17	Kimberley	39
18	Abdel	46
19	Bettine	50
20	Tersina	52
21	Daune	54
22	Augusta	59
23	Charissa	62
24	Nicola	69
25	Donella	72
26	Arlee	78
27	Corrie	79
28	Bartel	81
29	Cullen	83
30	Darin	86
31	Ashla	94
32	Hildegarde	96

10. DROP Table raw_materials;

```
DROP TABLE

Query returned successfully in 64 msec.
```

11. select * from toy_base;

	toy_id [PK] bigint	toy_name character (40)	toy_type bigint	age_group bigint	current_stock bigint	manufacturing_cost bigint	mrp bigint	profit bigint
1	1	Opossum, american virginia ...	6	8	49	64	144	80
2	2	Tamandua, southern ...	1	20	30	55	113	58
3	3	Black-footed ferret	6	16	51	70	114	44
4	4	Little brown dove	2	20	68	56	132	76
5	5	Porcupine, african	1	19	37	53	123	70
6	6	Armadillo, seven-banded ...	5	20	24	66	110	44

12. select name from employee order by joined_since limit 5;

	name character (200)
1	Florina ...
2	Sharia ...
3	Bettine ...
4	Hildagard ...
5	Donella ...

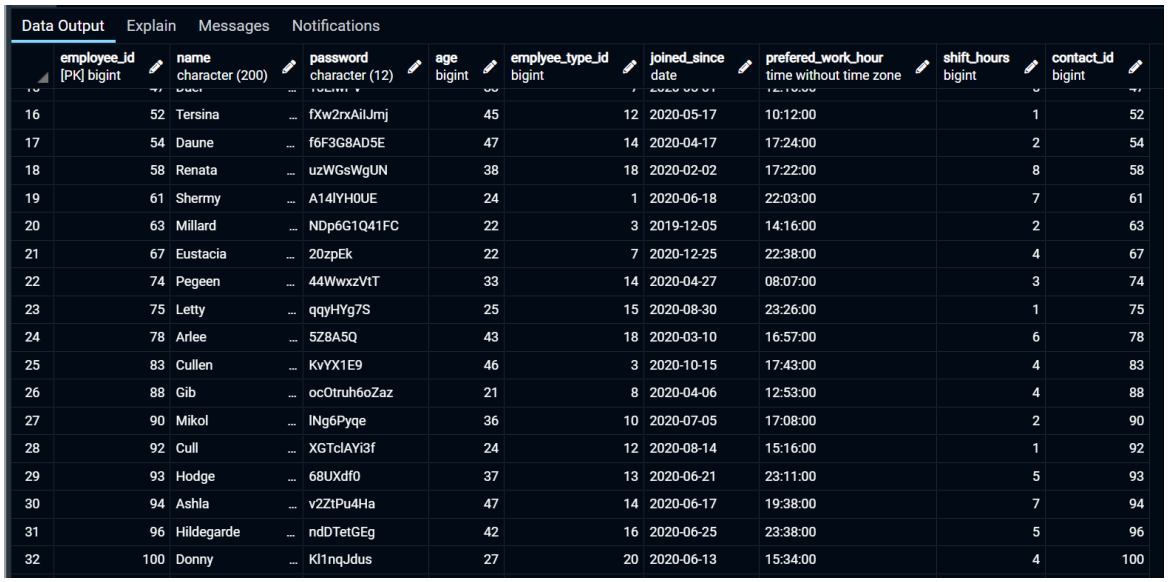
13. SELECT DISTINCT extract(year from joined_since) FROM employee

	extract numeric
1	2018
2	2019
3	2021
4	2020

14. select * from employee order by employee_id limit 25;

19	19	Marlie ...	58KWXwH7bwo	23	19	2019-11-03	15:17:00	6	19
20	20	Abbey ...	XjnJiFKV8	18	20	2021-10-20	11:10:00	5	20
21	21	Zared ...	RMjvrNBHs	21	1	2019-11-15	12:51:00	6	21
22	22	Sharia ...	HANiYWFqXACE	32	2	2018-12-15	14:03:00	5	22
23	23	Myrtice ...	QcLhq5KizJ	18	3	2020-03-18	09:24:00	2	23
24	24	Lisabeth ...	h5YYp3Tr6IGz	29	4	2019-04-29	13:20:00	6	24
25	25	Bord ...	Ec96h5wxNJ5	41	5	2021-04-22	23:46:00	2	25

15. SELECT * FROM employee WHERE joined_since BETWEEN '2019-12-01' AND '2021-01-01'



Data Output | Explain | Messages | Notifications

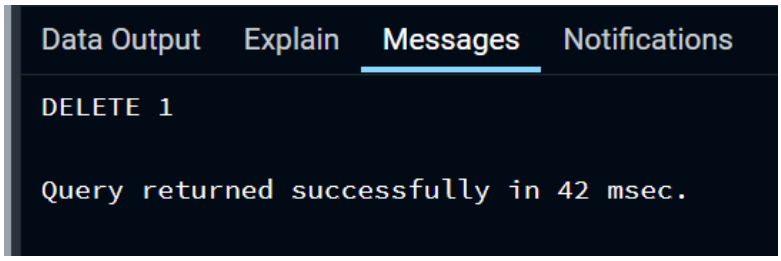
	employee_id [PK] bigint	name character (200)	password character (12)	age bigint	emplyee_type_id bigint	joined_since date	prefered_work_hour time without time zone	shift_hours bigint	contact_id bigint
16	52	Tersina	fXw2rxAiIJmj	45	12	2020-05-17	10:12:00	1	52
17	54	Daune	f6F3G8AD5E	47	14	2020-04-17	17:24:00	2	54
18	58	Renata	uzWGsWgUN	38	18	2020-02-02	17:22:00	8	58
19	61	Shermy	A14lYH0UE	24	1	2020-06-18	22:03:00	7	61
20	63	Millard	NDp6G1Q41FC	22	3	2019-12-05	14:16:00	2	63
21	67	Eustacia	20zpEk	22	7	2020-12-25	22:38:00	4	67
22	74	Pegeen	44WwxzVtT	33	14	2020-04-27	08:07:00	3	74
23	75	Letty	qqyHYg7S	25	15	2020-08-30	23:26:00	1	75
24	78	Arlee	5Z8A5Q	43	18	2020-03-10	16:57:00	6	78
25	83	Cullen	KvYX1E9	46	3	2020-10-15	17:43:00	4	83
26	88	Gib	ocOtruh6oZaz	21	8	2020-04-06	12:53:00	4	88
27	90	Mikol	lNg6Pyqe	36	10	2020-07-05	17:08:00	2	90
28	92	Cull	XGTclAYi3f	24	12	2020-08-14	15:16:00	1	92
29	93	Hodge	68UXdf0	37	13	2020-06-21	23:11:00	5	93
30	94	Ashla	v2ZtPu4Ha	47	14	2020-06-17	19:38:00	7	94
31	96	Hildegarde	ndDTetGEg	42	16	2020-06-25	23:38:00	5	96
32	100	Donny	Kl1nqJdus	27	20	2020-06-13	15:34:00	4	100

16. SELECT shop_order_id,shop_id,order_status AS Order_status FROM shop_orders;

Data Output | Explain | Messages | Notifications

	shop_order_id [PK] bigint	shop_id bigint	order_status boolean
59	59	54	true
60	60	5	false
61	61	20	true
62	62	52	false
63	63	39	true
64	64	5	true
65	65	38	true
66	66	25	false
67	67	12	false
68	68	53	false
69	69	53	false
70	70	45	false
71	71	50	true
72	72	60	false
73	73	54	false
74	74	18	true
75	75	47	false

17. DELETE FROM employee WHERE name = 'Any';



18. Calculates profit based on mrp and manufacturing cost

```
create or replace function profit_cal()
returns trigger AS
$$
begin
    update db.toy_base set profit = mrp - manufacturing_cost;
    return new;
end;
$$
language plpgsql;

drop trigger if exists profit_cal_trigger on db.toy_base;
create trigger profit_cal_trigger
after insert
on toy_base
for each row
execute procedure profit_cal();

insert into db.toy_base
(toy_id,toy_name,toy_type,age_group,current_stock,manufacturing_cost,mrp)
values (51,'mcqueen',2,16,10,65,100);
```

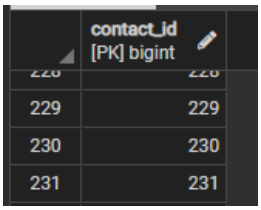
	toy_id [PK] bigint	toy_name character (40)	toy_type bigint	age_group bigint	current_stock bigint	manufacturing_cost bigint	mrp bigint	profit bigint
49	49	Porcupine, african	5	18	89	65	81	16
50	50	Capuchin, white-fronted	2	9	30	51	86	35
51	51	mcqueen	2	16	10	65	100	35

19. Create new contact id wherever a new user is created (employee, driver, material provider etc entities which uses contact table)

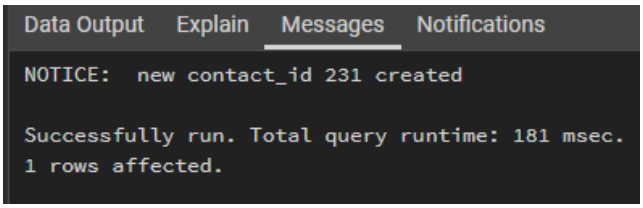
```
create or replace function new_contact()
returns integer as
$$
declare
temp_id bigint;
begin
    select max(contact_id) into temp_id  from db.contacts;
    temp_id := temp_id + 1;
    raise notice 'new contact_id % created',temp_id;
    insert into contacts(contact_id) values(temp_id);
    return temp_id;
end;
$$
```

```
language plpgsql;

select * from new_contact();
```



contact_id [PK] bigint	
229	229
230	230
231	231

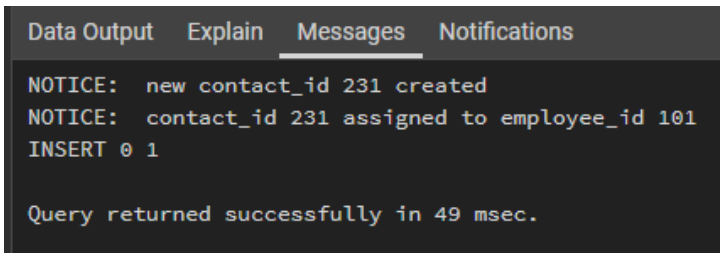


20. Creating trigger function insert contact_id in employee table.

```
create or replace function contact_id_insert()
returns trigger as
$$
declare temp_id integer;
declare max_id integer;
begin
    select into temp_id new_contact from new_contact();
    select max(employee_id) into max_id from employee;
    update employee set contact_id = temp_id where employee_id = max_id;
    raise notice 'contact_id % assigned to employee_id %', temp_id, max_id;
    return new;
end;
$$
language plpgsql;

drop trigger if exists contact_id_insert_trigger on employee;
create trigger contact_id_insert_trigger
after insert on db.employee
execute procedure contact_id_insert();

insert into employee (employee_id, name, password, age, emplyee_type_id,
joined_since, prefered_work_hour, shift_hours)
values (101, 'Jitanshu', 'qil3u4dy34', 20, 18, '2020-7-20', '8:00:00', 4);
```



	employee_id [PK] bigint	name character (200)	password character (12)	age bigint	emplyee_type_id bigint	joined_since date	prefered_work_hour time without time zone	shift_hours bigint	contact_id bigint
99	99	Kaleb ...	a5Fn52QA	25	19	2019-03-23	08:42:00	8	99
100	100	Donny ...	Kl1nqJdus	27	20	2020-06-13	15:34:00	4	100
101	101	Jitanshu ...	qil3u4dy34	20	18	2020-07-20	08:00:00	4	231

21. Creating trigger function insert contact_id in driver details table.

```
create or replace function driver_contact_insert()
returns trigger as
$$
declare temp_id integer;
declare max_id integer;
begin
        select into temp_id new_contact from new_contact();
        select max(driver_id) into max_id from driver_details;
        update driver_details set contact = temp_id where driver_id = max_id;
        raise notice 'contact_id % assigned to driver_id %', temp_id, max_id;
        return new;
end;
$$
language plpgsql;

drop trigger if exists driver_contact_insert_trigger on driver_details;
create trigger driver_contact_insert_trigger
after insert on db.driver_details
execute procedure driver_contact_insert();

insert into driver_details(driver_id) values(41);
```

Data Output | Explain | Messages | Notifications

```
NOTICE:  new contact_id 232 created
NOTICE:  contact_id 232 assigned to driver_id 41
INSERT 0 1

Query returned successfully in 59 msec.
```

Data Output | Explain | Messages | Notifications

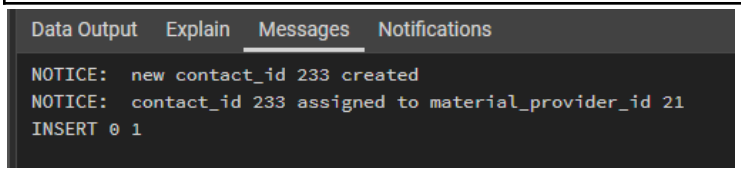
	driver_id [PK] bigint	driver_name character (20)	contact bigint	truck_no_plate bigint	truck_capacity bigint
39	39	Marian	169	7401	27
40	40	Leland	170	5571	39
41	41	[null]	232	[null]	[null]

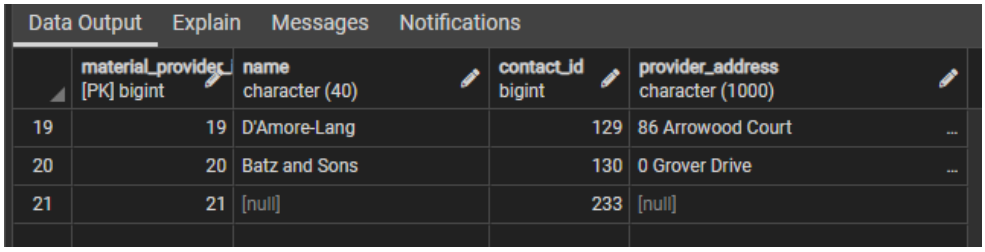
22. Creating trigger function insert contact_id in material providers table.

```
create or replace function mat_prov_contact_insert()
returns trigger as
$$
declare temp_id integer;
declare max_id integer;
begin
	select into temp_id new_contact from new_contact();
	select max(material_provider_id) into max_id from material_providers;
	update material_providers set contact_id = temp_id where
material_provider_id = max_id;
	raise notice 'contact_id % assigned to material_provider_id %', temp_id,
max_id;
	return new;
end;
$$
language plpgsql;

drop trigger if exists mat_prov_contact_insert_trigger on material_providers;
create trigger mat_prov_contact_insert_trigger
after insert on db.material_providers
execute procedure mat_prov_contact_insert();

insert into material_providers(material_provider_id) values(21);
```





23. Creating trigger function insert contact_id in Shop details table.

```
create or replace function shop_contact_insert()
returns trigger as
$$
declare temp_id integer;
declare max_id integer;
begin
```

```
        select into temp_id new_contact from new_contact();
        select max(shop_id) into max_id from shop_details;
        update shop_details set owner_contact = temp_id where shop_id =
max_id;
        raise notice 'contact_id % assigned to material_provider_id %', temp_id,
max_id;
        return new;
end;
$$
language plpgsql;

drop trigger if exists shop_contact_insert_trigger on shop_details;
create trigger shop_contact_insert_trigger
after insert on db.shop_details
execute procedure shop_contact_insert();

insert into shop_details(shop_id) values(61);
```

Data Output | Explain | Messages | Notifications

```
NOTICE:  new contact_id 234 created
NOTICE:  contact_id 234 assigned to material_provider_id 61
INSERT 0 1
```

	shop_id [PK] bigint	owner_name character (20)	shop_name character (20)	owner_contact bigint
60	60	Ruddy Scane	Temp	230
61	61	[null]	[null]	234

24. select driver_name from driver_details where driver_name like 'M%';

	driver_name character (20)
1	Mel
2	Mariquilla
3	Magdaia
4	Margarette
5	Mariejeanne
6	Marian

25. (Select * from employee order by age limit 5) union (select * from employee order by age desc limit 5) order by age;

Data Output | Explain | Messages | Notifications

	employee_id bigint	name character (200)	password character (12)	age bigint	emplyee_type_id bigint	joined_since date	prefered_work_hour time without time zone	shift_hours bigint	contact_id bigint
1	13	Catina ...	OkUY0dI	18	13	2019-07-20	13:18:00	7	13
2	20	Abbey ...	XjnJiFKV8	18	20	2021-10-20	11:10:00	5	20
3	26	Anastasie ...	dRdBsebGK	18	6	2021-02-12	17:28:00	5	26
4	23	Myrtice ...	QcLhq5KizJ	18	3	2020-03-18	09:24:00	2	23
5	4	Lyndel ...	FWQkDQW7wmS	18	4	2020-10-11	08:32:00	8	4
6	62	Charissa ...	Bq7xe634	49	2	2019-02-08	14:24:00	5	62
7	39	Kimberley ...	SMf1KL	49	19	2020-09-23	15:44:00	4	39
8	9	Ruby ...	jTB6BC	49	9	2020-02-17	20:58:00	4	9
9	16	Mufi ...	YHTtiOl6yASc	50	16	2019-04-07	15:39:00	3	16
10	32	Winn ...	w8IbxNF	50	12	2019-09-21	10:22:00	7	32

26. select department, salary_per_hour from employee_type order by salary_per_hour desc limit 5;

	department character (40)	salary_per_hour bigint
1	Product ...	95
2	Research and De...	93
3	Legal ...	93
4	Product ...	91
5	Human Recource...	89

27. select extract(year from employee.joined_since), avg(employee_type.salary_per_hour) from employee inner join employee_type on employee.emplyee_type_id = employee_type.employee_type_id group by extract(year from employee.joined_since);

	date_part double precision	avg numeric
1	2019	79.4687500000000000
2	2018	73.3333333333333333
3	2020	71.6451612903225806
4	2021	76.2812500000000000

28. Select name from employee left join employee_type on employee.employee_id = employee_type.employee_type_id where extract(year from joined_since) = 2020 and employee_type.salary_per_hour > 50

	name character (200)
1	Wileen
2	Lyndel
3	Leese
4	Ruby
5	Dudley
6	Loren
7	Ilario
8	Emery

29. select shop_details.shop_name, sales.sales_id from sales left join shop_orders on sales.order_id = shop_orders.shop_order_id left join shop_details on shop_orders.shop_id = shop_details.shop_id order by sales.sales_id limit 5;

	shop_name character (20)	sales_id bigint
1	Ventosanzap	1
2	Toughjoyfax	2
3	Transcof	3
4	Lotstring	4
5	Ventosanzap	5

30. select name, age, joined_since from employee;

	name character (200)	age bigint	joined_since date
1	Wileen	42	2020-11-04
2	Cully	28	2021-08-24
3	Corey	27	2021-07-21
4	Lyndel	18	2020-10-11
5	Janifer	46	2019-03-18
6	Elbert	42	2021-03-03

31. Select raw_materials.material_name, order_status from raw_material_orders left join raw_materials on raw_materials.raw_material_id = raw_material_orders.raw_material_id where order_status = True;

	material_name character (20)	order_status boolean
5	Stone	true
6	Plastic	true
7	Glass	true
8	Granite	true
9	Plexiglass	true
10	Glass	true
11	Granite	true
12	Brass	true

32. select driver_details.driver_name, est_arrival_indays from delivery left join driver_details on driver_details.driver_id = delivery.driver_id order by est_arrival_indays limit 5;

	driver_name character (20)	est_arrival_indays bigint
1	Edi	1
2	Ange	1
3	Valery	2
4	Cyril	2
5	Karel	3

33. select toy_base.toy_name, toy_base.current_stock, quantity from shop_orders join toy_base on shop_orders.toy_id = toy_base.toy_id limit 5;

	toy_name character (40)	current_stock bigint	quantity bigint
1	Sheathbill, snowy ...	81	55
2	Porcupine, african ...	89	12
3	Teal, hottentot ...	26	85
4	Pale white-eye ...	12	88
5	Downy woodpecker ...	81	96

34. select all_manufacturing_machines.machine_name,manufacturing_machines.last_service_date from manufacturing_machines left join all_manufacturing_machines on manufacturing_machines.machine_id = all_manufacturing_machines.machine_type_id where manufacturing_machines.last_service_date > 10;

	machine_name character (20)	last_service_date bigint
1	lafu-27130	17
2	covr-40291	31
3	hjng-08008	87
4	ksvy-68268	83
5	nudp-66292	17
6	ksxs-95576	74
7	glxu-24418	65

35. select count(employee_id), employee_type.department from employee left join employee_type on employee_type.employee_type_id = employee.emplyee_type_id group by employee_type.department;

	count bigint	department character (40)
1	11	Accounting ...
2	10	Product ...
3	10	Marketing ...
4	10	Legal
5	10	Support ...
6	10	Research and Develop...

36. select material_name, material_providers.name from raw_materials left join raw_material_orders on raw_materials.raw_material_id = raw_material_orders.raw_material_id left join material_providers on raw_material_orders.material_provider_id = material_providers.material_provider_id limit 5;

	material_name character (20)	name character (40)
1	Stone	Ryan-Hamill ...
2	Plexiglass	Zulauf, Orn and Parisian ...
3	Brass	Ledner-Turner ...
4	Rubber	Rau LLC ...
5	Brass	Ledner-Turner ...

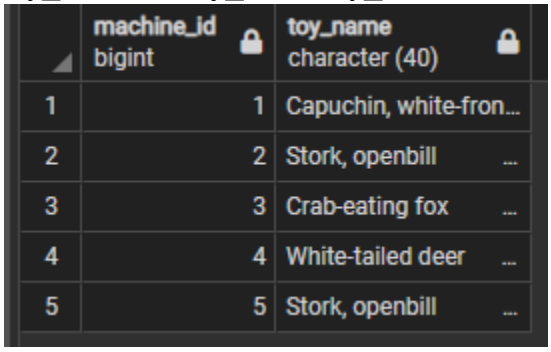
37. select shop_name, priority from sales left join shop_orders on shop_orders.shop_order_id = sales.order_id left join shop_details on shop_details.shop_id = shop_orders.shop_id order by priority desc;

	shop_name character (20)	priority double precision
1	Zathin	9.4
2	Duobam	9.2
3	Wrapsafe	8.4
4	Stronghold	8.3
5	Matsoft	8.3
6	Fix San	8.2
7	Zathin	8

38. select avg(mrp), type from toy_base left join toy_types on toy_types.toy_type_id = toy_base.toy_type group by toy_types.type;

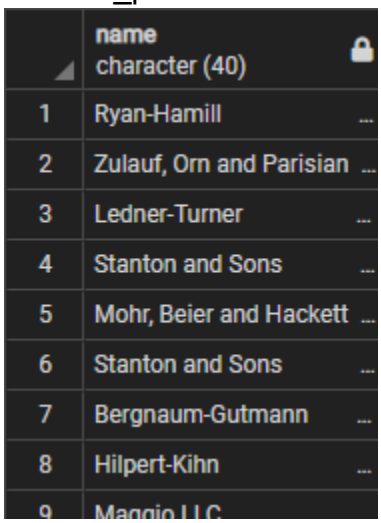
	avg numeric	type character (20)
1	108.3636363636363636	non
2	130.2222222222222222	tristique
3	108.6666666666666667	sit
4	111.8000000000000000	elementum
5	113.7500000000000000	donec
6	108.0000000000000000	nibh
7	120.0000000000000000	eget

39. select machine_id, toy_base.toy_name from manufacturing_machines left join toy_base on toy_base.toy_id = manufacturing_machines.toy_id limit 5;



	machine_id bigint	toy_name character (40)
1	1	Capuchin, white-fron...
2	2	Stork, openbill ...
3	3	Crab-eating fox ...
4	4	White-tailed deer ...
5	5	Stork, openbill ...

40. select name from material_providers right join raw_material_orders on raw_material_orders.material_provider_id = material_providers.material_provider_id where est_arrival_indays < 10;



	name character (40)
1	Ryan-Hamill ...
2	Zulauf, Orn and Parisian ...
3	Ledner-Turner ...
4	Stanton and Sons ...
5	Mohr, Beier and Hackett ...
6	Stanton and Sons ...
7	Bergnaum-Gutmann ...
8	Hilpert-Kihn ...
9	Maggio LLC