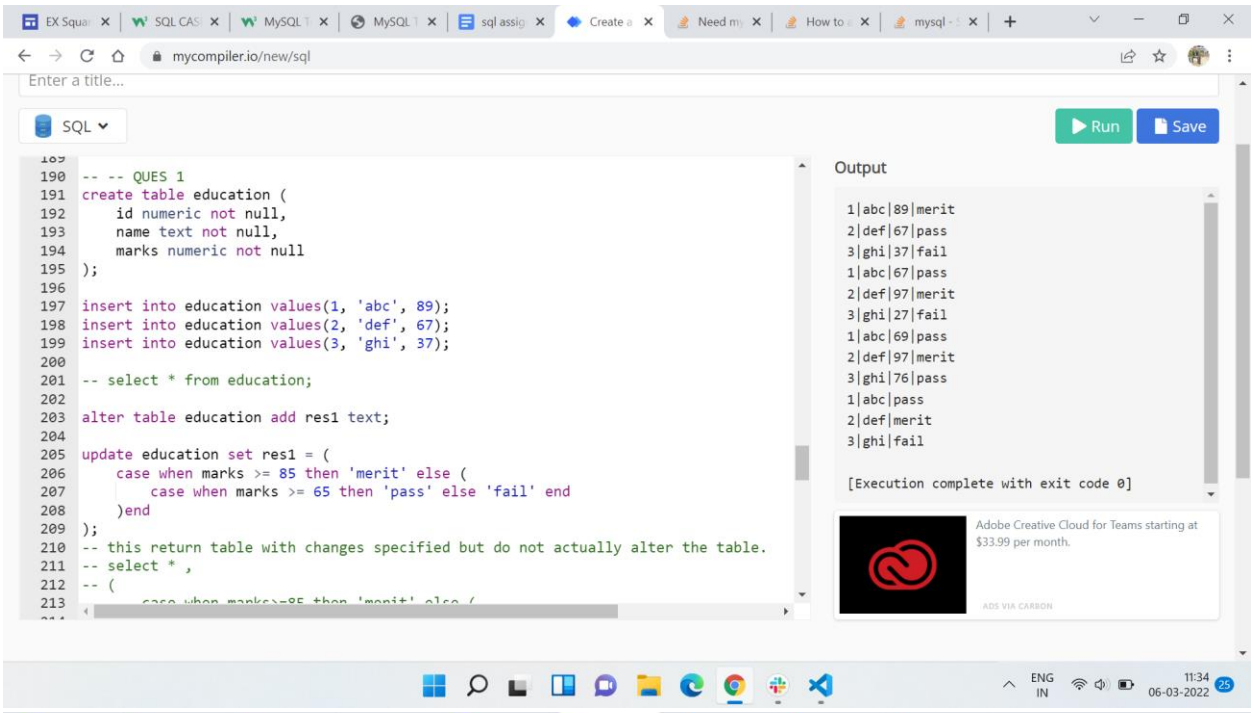
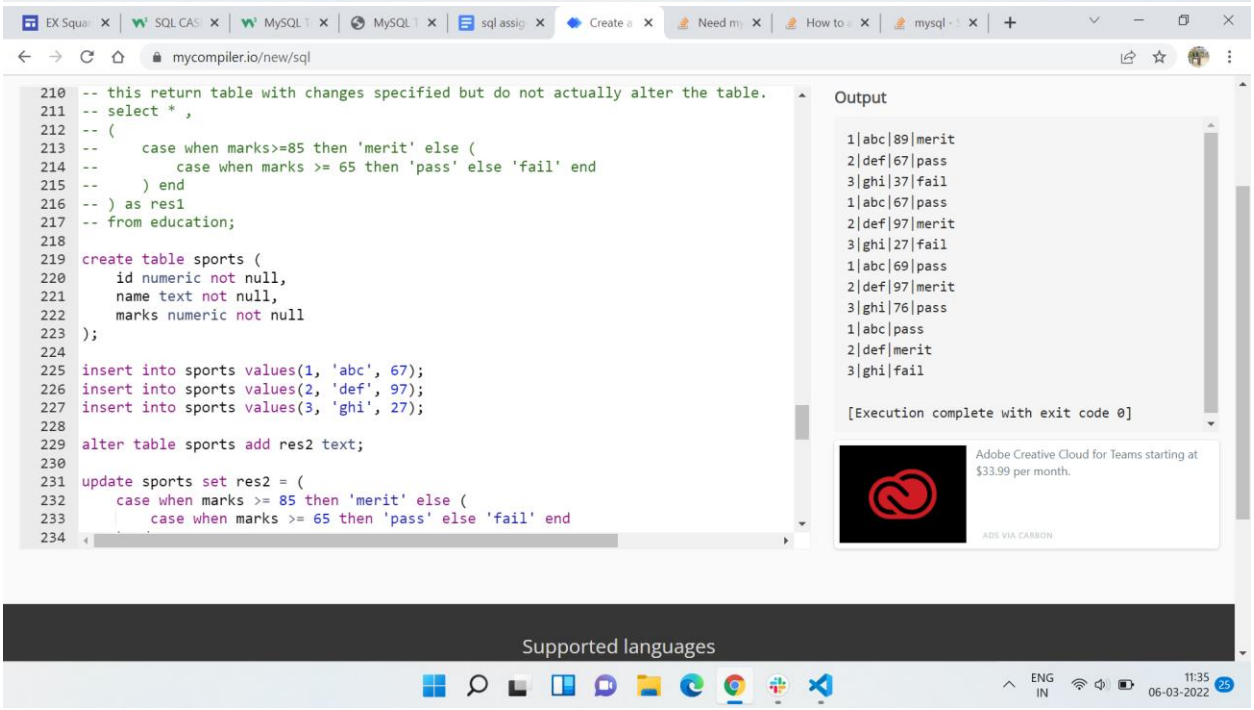
1.



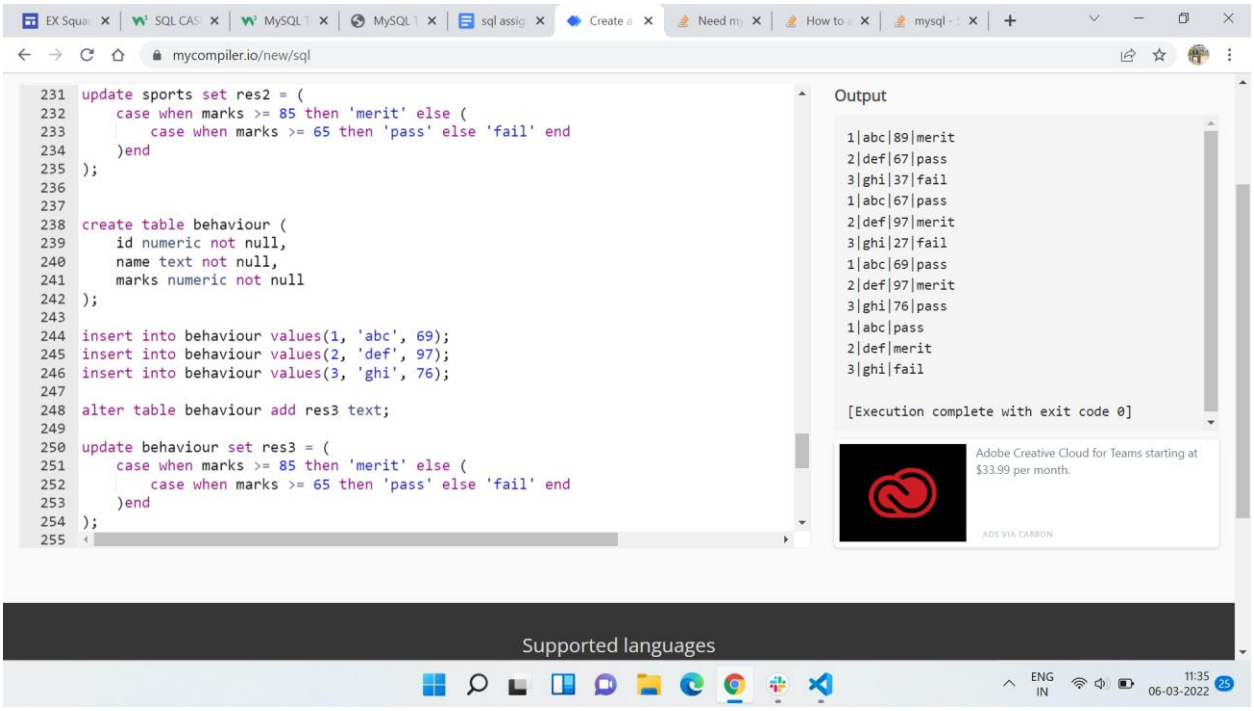
```
-- -- QUES 1
create table education (
    id numeric not null,
    name text not null,
    marks numeric not null
);

insert into education values(1, 'abc', 89);
insert into education values(2, 'def', 67);
insert into education values(3, 'ghi', 37);

-- select * from education;

alter table education add res1 text;

update education set res1 = (
    case when marks >= 85 then 'merit' else (
        case when marks >= 65 then 'pass' else 'fail' end
    )end
);
-- this return table with changes specified but do not actually alter the table.
-- select * ,
-- (
```

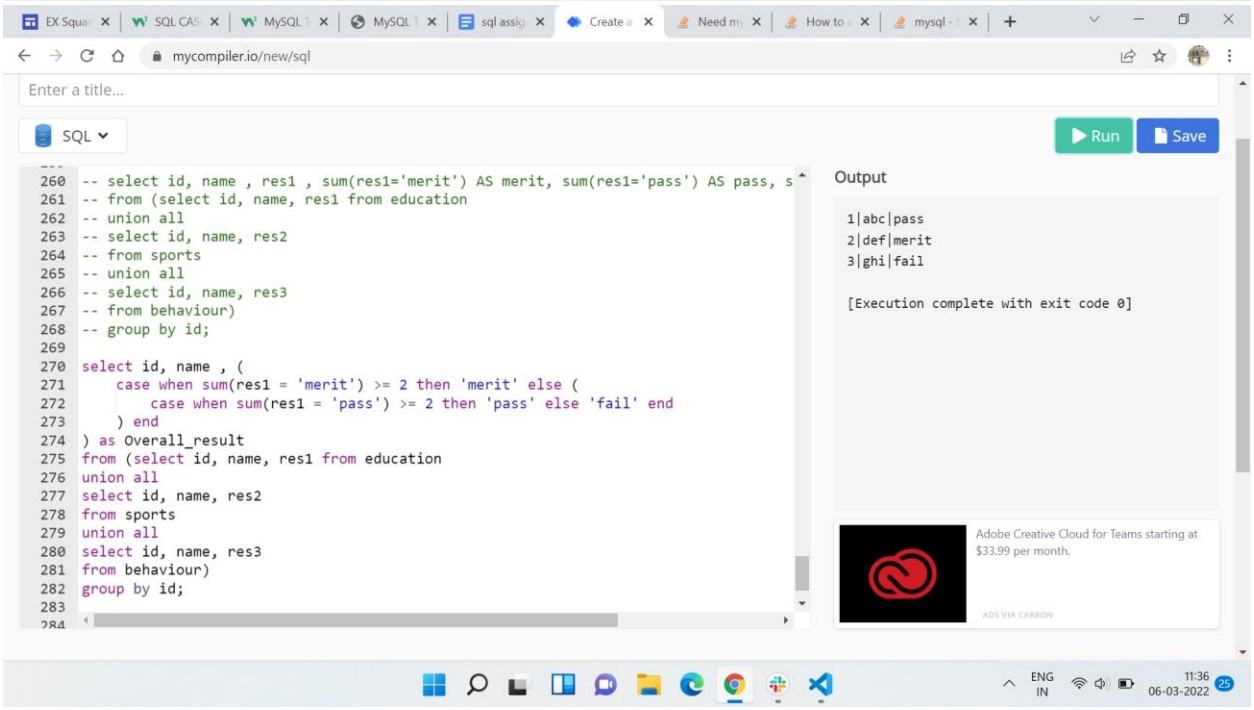
```
1|abc|89|merit
2|def|67|pass
3|ghi|37|fail
1|abc|67|pass
2|def|97|merit
3|ghi|27|fail
1|abc|69|pass
2|def|97|merit
3|ghi|76|pass
1|abc|pass
2|def|merit
3|ghi|fail

[Execution complete with exit code 0]
```



```
-- this return table with changes specified but do not actually alter the table.
-- select * ,
-- (
--     case when marks>=85 then 'merit' else (
--         case when marks >= 65 then 'pass' else 'fail' end
--     ) end
-- ) as res1
-- from education;

create table sports (
    id numeric not null,
    name text not null,
    marks numeric not null
);

insert into sports values(1, 'abc', 67);
insert into sports values(2, 'def', 97);
insert into sports values(3, 'ghi', 27);

alter table sports add res2 text;

update sports set res2 = (
    case when marks >= 85 then 'merit' else (
        case when marks >= 65 then 'pass' else 'fail' end
```

```
1|abc|89|merit
2|def|67|pass
3|ghi|37|fail
1|abc|67|pass
2|def|97|merit
3|ghi|27|fail
1|abc|69|pass
2|def|97|merit
3|ghi|76|pass
1|abc|pass
2|def|merit
3|ghi|fail

[Execution complete with exit code 0]
```


mycompiler.io/new/sql
update sports set res2 = (
case when marks >= 85 then 'merit' else (
case when marks >= 65 then 'pass' else 'fail' end
)end
);
create table behaviour (
id numeric not null,
name text not null,
marks numeric not null
);
insert into behaviour values(1, 'abc', 69);
insert into behaviour values(2, 'def', 97);
insert into behaviour values(3, 'ghi', 76);
alter table behaviour add res3 text;
update behaviour set res3 = (
case when marks >= 85 then 'merit' else (
case when marks >= 65 then 'pass' else 'fail' end
)end
);
Output
1|abc|89|merit
2|def|67|pass
3|ghi|37|fail
1|abc|67|pass
2|def|97|merit
3|ghi|27|fail
1|abc|69|pass
2|def|97|merit
3|ghi|76|pass
1|abc|pass
2|def|merit
3|ghi|fail
[Execution complete with exit code 0]
Adobe Creative Cloud for Teams starting at $33.99 per month.
ADS VIA CARBON
Supported languages
ENG IN
11:35
06-03-2022


mycompiler.io/new/sql
Enter a title...
SQL
Run
Save
-- select id, name , res1 , sum(res1='merit') AS merit, sum(res1='pass') AS pass, s
-- from (select id, name, res1 from education
-- union all
-- select id, name, res2
-- from sports
-- union all
-- select id, name, res3
-- from behaviour)
-- group by id;
select id, name , (
case when sum(res1 = 'merit') >= 2 then 'merit' else (
case when sum(res1 = 'pass') >= 2 then 'pass' else 'fail' end
) end
) as Overall_result
from (select id, name, res1 from education
union all
select id, name, res2
from sports
union all
select id, name, res3
from behaviour)
group by id;
Output
1|abc|pass
2|def|merit
3|ghi|fail
[Execution complete with exit code 0]
Adobe Creative Cloud for Teams starting at $33.99 per month.
ADS VIA CARBON
ENG IN
11:36
06-03-2022

2.

```sql
-- -- Inner Join
SELECT students.name, t2.Name
FROM students
inner join t2
on students.id = t2.id;
```

```sql
-- -- left join
SELECT students.name, t2.Name
FROM students
left join t2
on students.id = t2.id;
```

```sql
-- -- right join
SELECT students.name, t2.Name
FROM students
right join t2
on students.id = t2.id;
```

```sql
-- -- full join
SELECT students.name, t2.Name
FROM students
full join t2
on students.id = t2.id;
```

```sql
-- -- cross join
SELECT students.name, t2.Name
FROM students
cross join t2;
-- where students.id = t2.id;  including this it becomes a inner Join
```

```sql
-- -- union   // it do not allow duplicate values
select name from students
union
select Name from t2
```

```
-- union all // it allows duplicate values
select name from students
union all
select Name from t2
```

3.

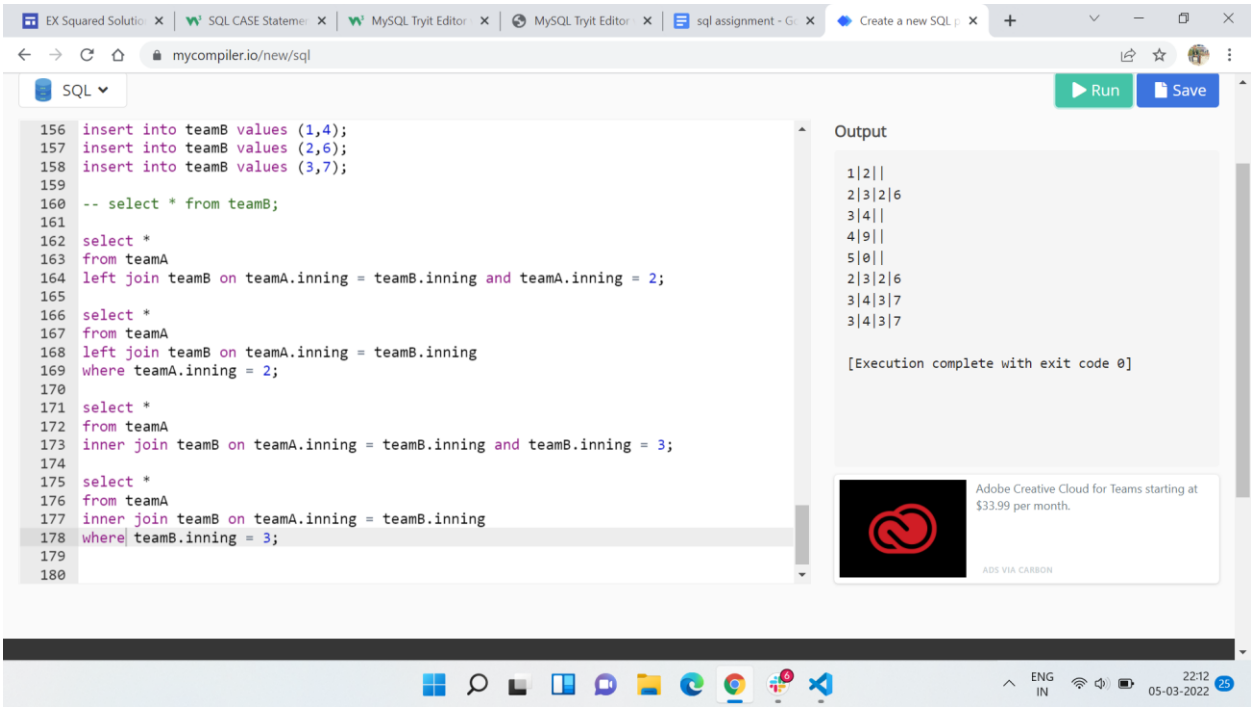
Ambiguous as the on property of join is trying to access the entire table.

The two inner select statements will run to give the table as output as a and b respectively. And then the entire content after the inner join will be returned based on the column a of table a and column b of table b.

4.

```
create table gender(
    name text not null,
    gend text not null
);

-- INSERT INTO students VALUES (1, 'X');
insert into gender values ('X', 'male');
insert into gender values ('Y', 'female');
insert into gender values ('Z', 'female');
insert into gender values ('A', 'male');

select * from gender;

update gender set gend = (
 case when gend = 'male' then 'female' else 'male' end
);

select * from gender;
```

5.



Yes there is a difference between query 1 and query 2 as in query 1 we are adding conditions on the join itself and in query 2 we are filtering out a result from the resulting join. And in this example the output is also being different.

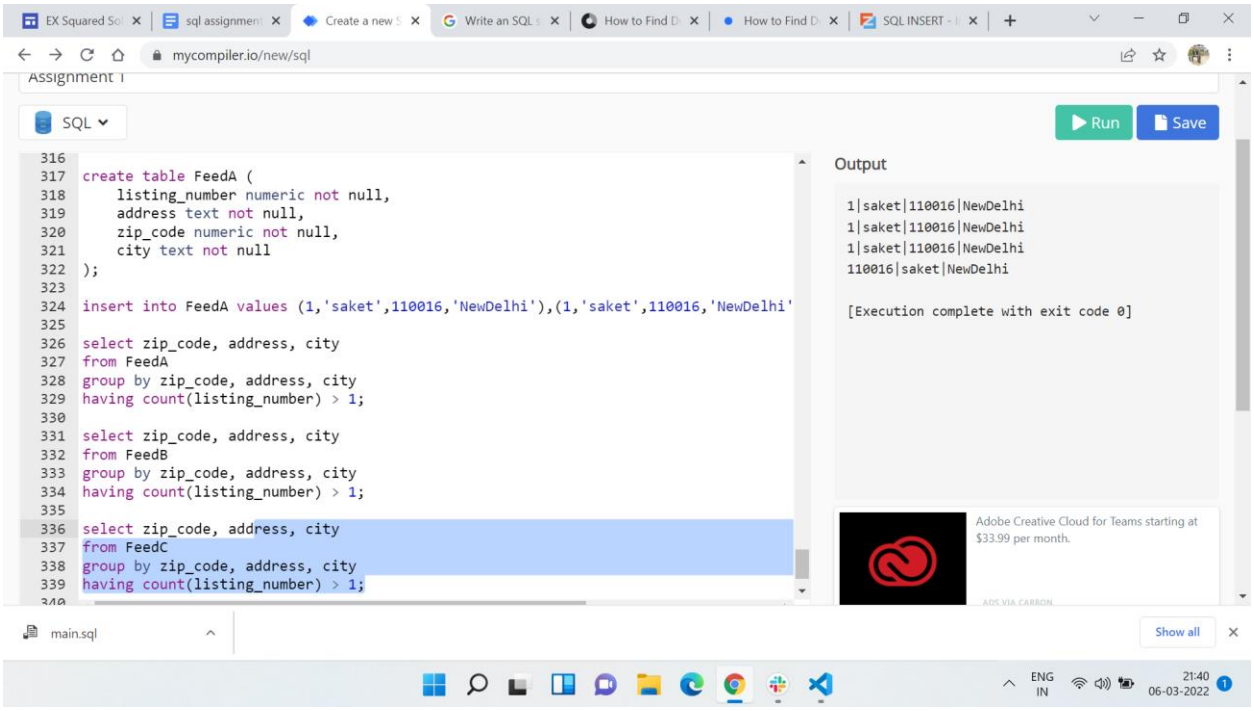
Yes there is a difference between query 3 and query 4 as in query 3 we are adding conditions on the join itself and in query 4 we are filtering out a result from the resulting join. But in this example the output comes out to be the same.

6.

```
select * from
t1 full join t2
on t1.item = t2.item;
```

7.

a)

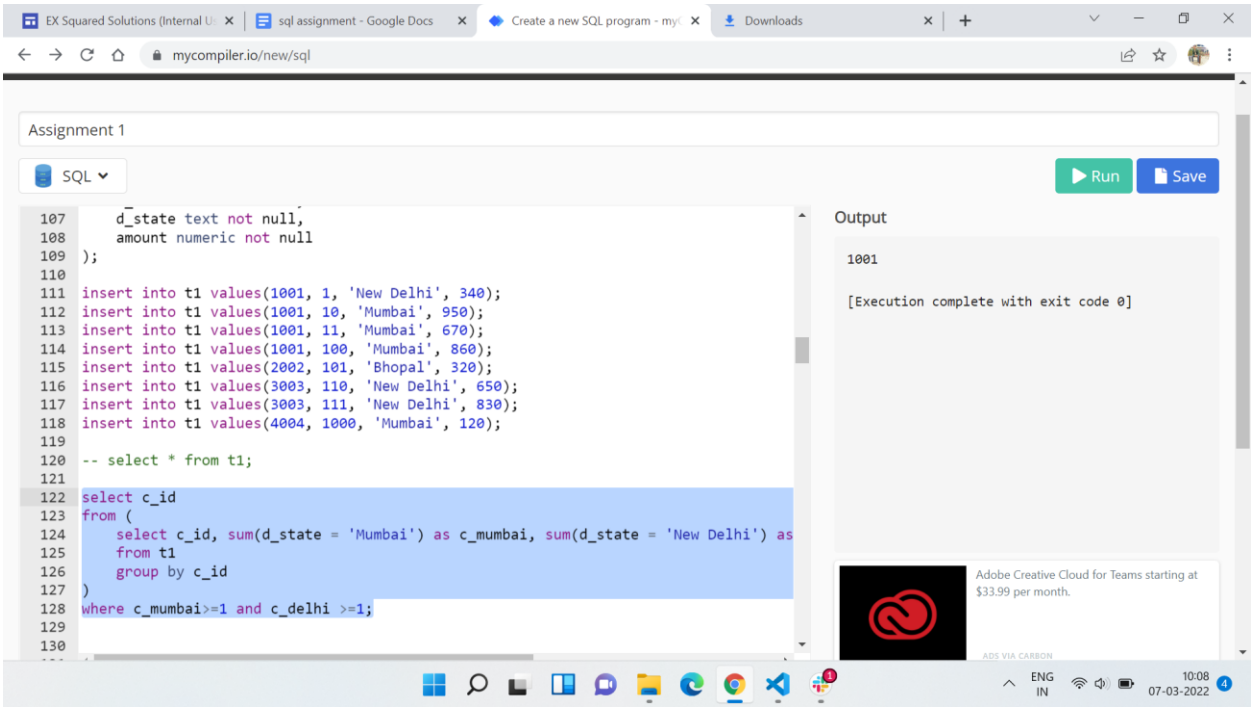



```
create table FeedA (
    listing_number numeric not null,
    address text not null,
    zip_code numeric not null,
    city text not null
);

insert into FeedA values (1,'saket',110016,'NewDelhi'),(1,'saket',110016,'NewDelhi'

select zip_code, address, city
from FeedA
group by zip_code, address, city
having count(listing_number) > 1;

select zip_code, address, city
from FeedB
group by zip_code, address, city
having count(listing_number) > 1;

select zip_code, address, city
from FeedC
group by zip_code, address, city
having count(listing_number) > 1;
```

b)


```
create table Final (
    listing_number numeric not null,
    address text not null,
    zip_code numeric not null,
    city text not null
);

insert into Final
select listing_number, address, zip_code, city
from (
    select * from FeedA
    union all
    select * from FeedB
    union all
    select * from FeedC
);

select * from Final;
```

8.



9.

```
select c_id
from (
    select c_id, sum(description = 'geologist') as geologist,
sum(description = 'astrogator') as astrogator,
sum(description = 'technician') as technician
from candidates
group by c_id
)
where geologist =1 and technician =1 and astrogator =1;
```

10.

```
create table design (
    workflow text not null,
    case1 numeric not null,
    case2 numeric not null,
    case3 numeric not null
);

insert into design values ('Alpha', 0 , 0 , 0);
insert into design values ('Bravo', 0 , 1 , 1);
insert into design values ('Charlie', 1 , 0 , 0);
insert into design values ('Delta', 0 , 0 , 0);

select * from design;

select workflow, case1+case2+case3 as Passed
from design;
```

11.