

Answer Paper
Alim Ul Karim

①

12. Apr. 2014

Started 6:20 pm
ended

MySQL select 'How you doing', 'Name Date' from 'Hello World';

① select [How you Doing], [Name Date]

from [Hello World];

| MSSQL, MS ACCESS,
ORACLE

② select AOID from A
where AOID IN (1, 3) AND
Allocation IN ('loc1', 'loc2');

④ select Concat (Allocation, bid) as display

from A;
results.

⇒ display

loc11

oc 2

oc 3

oc4

} MSSQL

Direct performing

string + number will
give error.

Allocation + str(bid) ⇒ possible

⇒ No error.

⑥ MySQL:
select Allocation + bid as display from A;

⇒ display

1
NULL

NULL

1

Oracle:

select Concat (Allocation, BID) as display from

A; display

Same as MSSQL output.

MySQL there is also Concat function. which give the
same result as MSSQL.

⑤ MS ACCESS:

Select Allocation + bid as display from A;

display

loc11

NULL

NULL

loc41

No error.

⑥ Select Allocation from A
where Allocation IN ('loc2', 'loc4');

⑦ Don't know the exact result.

For oracle: we could use concat function

Select Concat(Alocation, Bid) as display
from A;

Result should be same as MS SQL
answer.

Select Allocation + bid

Will result error and there is no
str() function to get int to string.
Number

(2)

8

MySQL

Display

NULL

NULL

0

NULL

NULL

MySQL/ACCESS/ORACLE

Display

NULL

NULL

$N1-V3 \neq N2-V3$

NULL

NULL

9

Select * From [X] where

3

Group by having Order

5

4

2

1

1, 3, 5, 4, 2 is the answer.

Ans (B)

10

IN query represents OR
AID IN (1, 2, 4) means

AID = 1 or

AID = 2 or

AID = 4

Difference between IN and OR is that
IN query executes as a block thus.

IN (1, 2, 4) =) (AID = 1 -- AID = 4)

As a result

answer is (C)

11

Select * from A
Where BID is not null AND
Allocation like 'OC %';

→ for some database wildcard *

Select * from A where BID is not null
AND Allocation like 'OC*';

12

Select * from A where Allocation NOT
like 'LOC%';

13

Select * from A where Allocation
& like 'LOC%';

14

Select * from A where Allocation NOT
like '%OC%';

↑
doesn't contain.

↑ wild ↑ wild
 └─ Card.

15

Clusted Index / Primary key is always sorted
to perform binary search.
Big O ($\log n$)

16

Non-primary key, Big O (n)

17

Non clustered indexes are those index which are made against other columns in database based on clustered index or primary key.

PK or clustered index

ID	FName
1	Zakir
2	Mahadi
3	Azhar
4	Alice

Primary key index or clustered index.

If a non clustered index is made against pk for FName then in background ~~it also be~~ 0

Non clustered index

PK or clustered index

FName	ID
Alice	4
Azhar	3
Mahadi	2
Zakir	1

Now if we search against FName or ~~ID~~ ID in this table

$O(\log n)$

Creating takes a sorting.

So creating or modifying $O(n \log n)$
 \uparrow
 best sorting

Making non-cluster index come at cost so don't make every one of the columns based.

~~only do this when we are constantly searching with it~~

for Non-clustered indexes there is always 2 entries.

18 Answered in details in (17)

$O(n \log n)$

Don't make every column non-clustered index.

19 Number/oid performs better than Varchar

Think ~~through~~ through

1 varchar takes 8 bits of space
so if you want primary key of
4 characters it would be $8 \times 4 = 32 \text{ bits}$

You can combine character with number
but if you store serial numbers as
varchar it ~~will~~ will overrun

int size after keeping only 4
chars. On the other hand if

you have int ^{32 bits}, you can have

(-2 billion to 2 billion) 4 billion possible serials. Which
~~much~~ much more ~~opt~~ optimize and better
solution. Another thing about int that
numbers perform better in terms of sorting
and binary searching.

20

nvarchar(8) or char(8)

Always take full space

both
take
fixed
space

$\left\{ \begin{array}{l} \text{varchar}(8) \rightarrow \text{Trimmed space / Not all characters filled} \\ \text{char}(8) \rightarrow \text{All characters must be filled} \end{array} \right.$

varchar(8) / nvarchar(8)

space ←
8x8 = 64 bits

"hello"

space ←
8x8 = 64 bits

char(8) / nchar(8)

"hello"
 ↓ ↓ ↓
 space space

even though you kept just 'hello' but
in two formats it will differ.

To get the hello from nchar/char.

SQL ⇒ Select * from T1 where n1 = "hello...";
 or
 SQL ⇒ Select * from T1 where n1 like 'hello%';

21 Between query varies from database to database. But here database is given

MS SQL SERVER:

Test \geq Can1 AND Test \leq Can2

So ①, ④ would be the answer.

22 SQL \Rightarrow Select * from Employee where
birthdate Not Between '#12-Apr-1960'
AND '#12-Apr-2014' #'

\Rightarrow Since between acts differently on database to database it is best not to use it.

So above query can also be written as.

SQL \Rightarrow Select * from Employee
where ~~Not~~ (Birth date \geq '12-Apr-1960'

AND Birthdate \leq '12-Apr-2014');

Some databases like MS Access Required to
have # Date #

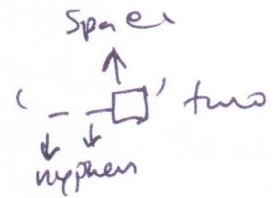
↑ ↑
before end a date.

However, Oracle, MySQL, MS SQL doesn't require
sign.

5

23

MySQL, MSSQL, Oracle.
General comments are written
hyphens. and one space.



-- Example of one line comment

MSSQL, MySQL, Oracle
/* multiline comments */

24

- (E) incorrect because there is no ID field in B table.
- (F) incorrect same reason.
- (G) "
- (H) incorrect, C table doesn't have BID.

29

Select ^{top 5} DepartmentID, Avg (Salary), DepartmentName
from PersonX
Group By DepartmentID, DepartmentName
Having Avg (Salary) > 5000 ;

for unique column.

Select Distinct AOLD from A

where $AOID > 0$ AND $AOID < -2$ AND
 $AOID > -5$ AND
 $(AOID + AOID) = (AOID * AOID) ;$

MySQL

Declare variable :

SELECT @var := 'value';

SELECT @var := 'value',
Select @substr = substr('hello', 1, 3);

-- ④ substr = $\begin{matrix} \text{hel} \\ 1\ 2\ 3 \\ \uparrow \\ \text{start} \end{matrix}$

select @ substr:2 ^{start} substring('hello', 2, 3);

-- substring = ell
3 len.

MSQL

MySQL select var 2 substring('hello', 2, 3);

-- var = ell

```
-- var = ell
-- select var = substring (field, starting = 1, len)
```

For this question Answer:

For this question Answer:
 MSSQL> select ~~substring~~ concat(substring(Abention, 2, 2),
 BID) as Display
 from A;

6

Oracle

Select Concat (Allocation substring (Allocation, 2, 2), BID) AS
Display From A;

MySQL / MSSQL → same

Select Concat (substring (Allocation, 2, 2), BID) AS
display FROM A;

Advance 60 Marks

1

- (i) (a) intersect
(b) join
(c) inner join } inner join and join represents same thing.

(2) (i) L NOT R

- (3) (e) left join
(f) left outer join } something.

- (4) (g) Right join
(h) Right outer join } same thing.

(5) (i) R NOT L

(6) full outer join

- (7) (a) intersect
AND also full outer join.

②

join always make cartesian product.

$$\{a, b\} \times \{c, d, e\} = \{(a, c), (a, d), (a, e), (b, c), (b, d), (b, e)\}$$

Select AID, B.AID, BID, AName, BName.

(i) Full outer join
from A full outer join B on (A.AID = B.AID)

Full outer join gives both side tables all rows and where not matched cell is null.

AID	AID	BID	AName
	1	1	A1
	2	N	A2
	3	N	A3
	4	1	A4

X

BID	BName
1	B1
2	B2
3	B3
4	B4
5	B5

=>

AID	BID	AName	BName	BID
1	1	A1	B1	1
1	1	A1	B2	2
1	1	A1	B3	3
1	1	A1	B4	4
1	1	A1	B5	5
2	N	A2	B1	1
2	N	A2	B2	2
2	N	A2	B3	3
2	N	A2	B4	4
2	N	A2	B5	5
3	N	A3	B1	1
3	N	A3	B2	2
3	N	A3	B3	3
3	N	A3	B4	4
3	N	A3	B5	5
4	1	A4	B1	1
4	1	A4	B2	2
4	1	A4	B3	3
4	1	A4	B4	4
4	1	A4	B5	5

Not exact representation

So where n we use
A.AID = B.AID

Only Red ones Selected like this.

when Full join
It includes all the rows from both tables at least once.

7

So for

full join

(1) Answer is result 3 (C)

Result 1 has all B table but not A and other do not satisfy.

(2) Left ~~outer~~ outer on Left join are same thing.

Left outer means all rows from table ON left must appear. based on the given condition.

⇒ FROM A Left JOIN B (condition)
 ↑ ↑
 left Right

~~So the result is so should be~~

Here, in left table BID = {1}.

~~So~~ only BID = 1 from B table should be included. and since left join ^{given} all rows from A table should be there.

Finally the result ~~show~~ should be (B) or Result 2.

Left join

(iii) Right outer / Right join same thing.

$A.AID = B.BAID$

Right so all rows from B table should be there.

IN B table $AID = \{1, 2, 3\}$.

So the result should only contain data from

$AID = 1, 2, 3$. ~~and others will be~~

Finally the Result is Result 1 or (A).
Right join

(iv) Answered in (ii)

(v) Answered in (iii)

(vi) Inner join / join / where $A.AID = B.BID = A.BID$

same Representation.

In inner join only shows both sides matched rows only.

Clearly the result is 4 or D.

inner join.

3.

select UPPER (Field) ;
Makes char Upper

Calculating Date difference:

MySQL:
To get only ~~the~~ Date difference. there is function
DateDiff (startDate, endDate) -- result in days diff.

Another one is
TimeStampDiff (whichOne, startDate, endDate)
↓
Day
Months
Second
Year
Minute
:

get current date &
time = Now() ;
↓
function.

MSSQL
get current date time = getDate()
↓
function.

DateDiff (whichOne, startDate, endDate)
=> equivalent to MySQL TimeStampDiff


```

SELECT
  Concat( Replace(UPPER(FirstName), 'UL', '_'), Space ' ',
    LastName, ' ', '(',
    DateDiff(year, Birthdate, GetDate()), ' ', ')',
    DateDiff(year, BirthDate, GetDate()) As Age
FROM Employee
Where Age >= 20
ORDER BY Age Desc ;

```

4
= (a) syntax error on dbo.Patients R.

(b) correct.

(c) like doesn't satisfies.

(d) AuthorName and <= is wrong.

(e) AuthorName should be in like.

5

(a) Select ~~keep~~(Rate) missing

x ORDER syntax is wrong

(b) IN query doesn't have maths,
Count(Author ID) should be > 2 not 3.

x Group by book Name missing

(c) s. subject OR should be in a parenthesis.

x Count(Author ID) > 2 missing.

(d) Syntax Correct and satisfies all the condition.
IN this case Left join or join will not
give different result because final results
will be determinate by the where and
having query.

Even though Left gives all the books
but where and having selects only
specific ones.

(e) Correct, as (d)

✓

(6) Since there is a Not, it will give the basic inner join result.

Always take the small field to determine result. ↓

Results:

Always take smaller: B table AID = {1, 2, 3}

AID	B table AID	AName	BName
1	1	A1	B1
2	2	A 2	B2
2	2	A 2	B3
3	3	A 3	B5
1	1	A1	B4

⑦ // Correct SQL:
 Select A.AID, B.BID, AName, BName
 From A join B on (A.BID = B.BID)

If alias is declared then in the join condition it must be called with that alias.

Inner join (A.BID = B.BID)

page 7, Ques 2, Result 4 (D).

8// correct SQL: Select A.AID, B.AID, AName, BName
from A join B ON (A.BID <> B.BID)

AID	AID	AName	BName
+	A +	A	
1	2	A1	B2
1	2	A1	B3
1	3	A1	B5
2	1	A2	B1
2	1	A2	B4
2	3	A2	B5
3	1	A3	B1
3	2	A3	B2
3	2	A3	B3
3	1	A3	B4
4	1	A4	B1
4	2	A4	B2
4	2	A4	B3
4	1	A4	B4
4	3	A4	B5

described
in the
full outer join.

A x B
Result
without
Matching.

(9)

Correct SQL: Select A.AID, B.AID, AName, BName.
from A join B ON NOT

(A.BID <> B.BID)

Query will return result exactly same as

(7)

'Not' will reverse it.

(10)

two months ago
haven't rated
have 4 authors.
Authors write one book in past

MySQL ↘

SQL:

Select
BookName, AuthorName, Published Date, Count(Rate) as
from book b join Author A ON (A.BookID = b.bookID)
join [Rating] R ON (R.BookID = b.bookID)
Where DateDiff(month, publishedDate, GetDate()) = 2

AND A.AuthorID

IN (select AuthorID from Author
bookID <> b.bookID)

Where
Group
Having

by bookName, AuthorName, AuthorID
Count(AuthorID) > 4 AND
SUM(R.Rate) > 0 ;

Rate

11

Only based on given Data in table A
page 2.

11

- A. Same
- B. Same
- C. Same
- D. NOT Same.
- E. NOT Same
- F. NOT Same, may return more than 2.
- G. Same.
- H. Same.

12

ASC desc top 1 2 3
ASC top 1 = 1

No syntax Error Based
on MSSQL.

Case 1

Result
3 = Find It

Case 2

Result
1 = Find One

Case 3

Result
Not Found

