# Oracle SQL & PL/SQL

SQL, SQL Server, Tutorials, Oracle, PL/SQL, Interview Questions & Answers, Joins, Multiple Choice Questions, Quiz, Stored Procedures, Select, Insert, Update, Delete and other latest topics on SQL, SQL Server and Oracle.

RSS Feed

## SQL JOIN Interview Questions



These are three examples of SQL Join Interview Questions:

**Example 1:**
Below are three tables: Client, Bank and Bill. The question is based on these three tables.

Creating tables:

```
CREATE TABLE Client
(
ClientID int,
ClientName varchar(255),
Primary Key (ClientID)
);


CREATE TABLE Bank
(
BranchID int,
BranchName varchar(255),
ClientID int,
Primary Key (BranchID),
FOREIGN KEY (ClientID) REFERENCES Client (ClientID)
);
```

### SQL Server - Multiple Choice Questions

- SQL - Multiple Choice Questions

### SQL Server, Oracle, SQL & PL/SQL

If you like reading this SQL Server, Oracle, SQL / PLSQL blog, please help increase the blogs' visibility by clicking on the 'g +1' button.

### SQL Server / Oracle SQL

- SQL Introduction
- SQL Syntax
- SQL Select
- SQL Insert
- SQL Update
- SQL Delete
- SQL Joins
- SQL Examples of Joins
- SQL Explicit vs. Implicit Joins
- SQL Group By
- SQL Group By Examples
- SQL Having
- SQL - IN
- SQL - NULLs
- Functional Dependencies
- Normalization
- ACID Properties
- SQL SubQueries
- SQL - Queries With Examples
- SQL Views
- Insert, Update, Delete Views
- SQL Join Views
- SQL Inline Views
- SQL - Nth Highest Salary
- SQL Second Highest Salary
- SQL - Difference Truncate / Delete

### Oracle PL/SQL

- PL/SQL Introduction
- PL/SQL Procedures
- PL/SQL Functions
- PL/SQL Collections
- PL/SQL Records
- PL/SQL Table Based Records
- PL/SQL Programmer Defined Records
- PL/SQL Cursor Based Records
- PL/SQL Tables
- PL/SQL Varrays
- PL/SQL Nested Tables
- PL/SQL Loops
- PL/SQL Triggers
- PL/SQL Cursors
- PL/SQL Implicit Cursors
- PL/SQL Explicit Cursors
- PL/SQL REF Cursors
- PL/SQL Cursor For Loop
- PL/SQL Cursors with Parameters
- PL/SQL Where Current Of and For Update
- Examples of PL/SQL Cursors
- PL/SQL Exceptions
- PL/SQL Interview Questions
- SQL PL/SQL Sitemap

```
CREATE TABLE Bill
(
InvoiceID int,
Year DateTime,
BranchID int,
Amount int,
Primary Key (InvoiceID),
FOREIGN KEY (BranchID) REFERENCES Bank(BranchID)
);
```

Inserting data into tables:

```
insert into Client values (1, 'O_A')
insert into Client values (2, 'O_B')
insert into Client values (3, 'O_C')
insert into Client values (4, 'O_D')
insert into Client values (5, 'O_E')

insert into Bank values (1, 'B_1', 1)
insert into Bank values (2, 'B_2', 2)
insert into Bank values (3, 'B_3', 3)
insert into Bank values (4, 'B_4', 4)
insert into Bank values (5, 'B_5', 5)
```



```
insert into Bill values (1, '2012-01-01 00:00:00.000',    1,    100)
insert into Bill values (2, '2013-01-01 00:00:00.000',    2,    200)
insert into Bill values (3, '2012-01-01 00:00:00.000',    3,    300)
insert into Bill values (4, '2017-01-01 00:00:00.000',    4,    400)
insert into Bill values (5, '2012-01-01 00:00:00.000',    5,    500)
insert into Bill values (6, '2012-01-01 00:00:00.000',    1,    900)
insert into Bill values (7, '2013-01-01 00:00:00.000',    1,    900)
```

Verifying data in the tables:

Select * from Client

| ClientID | ClientName |
|---|---|
| 1 | O_A |
| 2 | O_B |
| 3 | O_C |



## SQL Tutorials / Topics

| | |
|---|---|
| 4 | O_D |
| 5 | O_E |

Select * from Bank

| BranchID | BranchName | ClientID |
|---|---|---|
| 1 | B_1 | 1 |
| 2 | B_2 | 2 |
| 3 | B_3 | 3 |
| 4 | B_4 | 4 |
| 5 | B_5 | 5 |

Select * from Bill

| InvoiceID | Year | BranchID | Amount |
|---|---|---|---|
| 1 | 2012-01-01 00:00:00.000 | 1 | 100 |
| 2 | 2012-01-01 00:00:00.000 | 2 | 200 |
| 3 | 2012-01-01 00:00:00.000 | 3 | 300 |
| 4 | 2012-01-01 00:00:00.000 | 4 | 400 |
| 5 | 2012-01-01 00:00:00.000 | 5 | 500 |
| 6 | 2012-01-01 00:00:00.000 | 1 | 900 |
| 7 | 2012-01-01 00:00:00.000 | 1 | 900 |

**Question:**
Retrieve all invoices from table bill for year 2012 and 2013 which belong to client 'O_A':

```
Select i.InvoiceID,i.YEAR, i.BranchID, i.Amount from Bill as i
join Bank as b on i.BranchID = b.BranchID
join Client as o on o.ClientID = b.ClientID
where i.Year >= '2012-01-01' and i.Year <= '2013-01-01'
and o.ClientName = 'O_A'
```

**Result:**

| | InvoiceID | YEAR | BranchID | Amount |
|---|---|---|---|---|
| 1 | 1 | 2012-01-01 00:00:00.000 | 1 | 100 |
| 2 | 6 | 2012-01-01 00:00:00.000 | 1 | 900 |
| 3 | 7 | 2013-01-01 00:00:00.000 | 1 | 900 |

**Example 2:**

Below are three tables: salesman, Client and Items. The questions below are based on these three tables.

Creating tables:

```
CREATE TABLE salesman
(
salesman_id int,
Name varchar(255),
Age int,
Salary int,
Primary key (salesman_id)
);

CREATE TABLE Client
(
Cust_ID int,
```

by
- SQL - Diff SCOPE_IDENTITY / IDENT_CURRENT / @@IDENTITY
- SQL Coalesce
- SQL - Import Data into SQL Server Table
- SQL - Delete Duplicate Records in SQL Server Table
- SQL - Check If a Column Exists in SQL Server Table
- SQL - Has_perms_by_name
- SQL - Difference between Logins / Users
- SQL - Diff between Table Variable/Temp Table
- SQL - SQLServer 2012 features
- SQL - SERVERPROPERTY
- SQL - Find Primary Key / Foreign Key in Table
- SQL - Sequence
- SQL - Columnstore Indexes
- SQL - Difference Primary / Candidate Key
- SQL - Throw - SQL Server 2012
- SQL - Difference GUID / INT
- SQL Server 2012 Enhancements
- SQL - Diff between Cross Join / Full Outer Join
- SQL - NTILE()
- SQL - Object Dependencies
- SQL - Email Reminder on Updating Table
- SQL - Object Definition
- SQL Server - Report Builder 3
- SQL Server - User Defined Roles
- SQL Server - Filetables
- SQL Server - Contained Databases
- SQL Server - Update from Select
- SQL Server - XML Methods
- SQL Server - Add a Column to Table
- SQL Server - Insert Multiple Rows
- SQL Server - Add Identity Column to Table
- SQL Server - Indexes
- SQL Server - Diff Functions / Stored Procedures
- SQL - Comparison Operators Any / All
- SQL - Difference Primary / Foreign Key
- SQL - Difference Replace / Stuff
- SQL - ROW_NUMBER()
- SQL - Intersect
- SQL Server - Local Variables
- SQL Server - Update with Join
- SQL Server - Difference Merge / Hash Join
- SQL Server - Difference Update / Exclusive Lock
- SQL Server - Difference Instance / Database
- SQL Server - Difference Datetime2 / Datetime
- SQL Server - Difference Disable / Drop Indexes
- SQL Server - Difference Decimal / Float
- SQL Server - DATEDIFF with Examples
- SQL Server - Difference

```
Name varchar(255),
City varchar(255),
IndustryType char,
Primary key (Cust_ID)
);
```



Raiserror / Throw
- SQL Server - Check If Column Exists in Table
- SQL Server - Query Excel Using Linked Server
- SQL Server - WaitFor
- SQL Server - Difference Rank, Dense_Rank & Row_Number
- SQL Server - PATINDEX Examples
- SQL Server - COLLATE Examples
- SQL Server - Difference CHAR / VARCHAR
- SQL Server - Interview Questions for Testers
- SQL Interview Questions
- Difference - GRANT / DENY / REVOKE
- Subqueries in WHERE clause Examples
- How to check indexes on a table?
- DATENAME examples
- SQL Join Interview Questions

```
CREATE TABLE Items
(
Number int,
Order_date date,
Cust_ID int,
salesman_id int,
Amount int,
Foreign Key (Cust_ID) references Client (Cust_ID),
Foreign Key (salesman_id) references salesman (salesman_id)
);
```

Inserting data into tables:

```
Insert into salesman values (1, 'Amir', 61, 140000)
Insert into salesman values (2, 'Balbir', 34, 44000)
Insert into salesman values (5, 'Chander', 34, 40000)
Insert into salesman values (7, 'Damdar', 41, 52000)
Insert into salesman values (8, 'Kumar', 57, 115000)
Insert into salesman values (11, 'Jaggu', 38, 38000)

Insert into Client values (4, 'Samsung','Delhi', 'J')
Insert into Client values (6, 'Panasonic','Orange', 'J')
Insert into Client values (7, 'Nokia','Jamshedpur', 'B')
Insert into Client values (9, 'Apple','Jamshedpur', 'B')

Insert into Items values (10, '8/2/97', 4, 2, 540)
Insert into Items values (20, '1/30/96', 4, 8, 1800)
Insert into Items values (30, '7/14/94', 9, 1, 460)
Insert into Items values (40, '1/29/95', 7, 2, 2400)
Insert into Items values (50, '2/3/95', 6, 7, 600)
Insert into Items values (60, '3/2/95', 6, 7, 720)
Insert into Items values (70, '5/6/95', 9, 7, 150)
```

Verifying data in the tables:

Select * from Items

| Number | Order_date | Cust_ID | salesman_id | Amount |
|---|---|---|---|---|
| 10 | 1997-08-02 | 4 | 2 | 540 |
| 20 | 1996-01-30 | 4 | 8 | 1800 |
| 30 | 1994-07-14 | 9 | 1 | 460 |
| 40 | 1995-01-29 | 7 | 2 | 2400 |
| 50 | 1995-02-03 | 6 | 7 | 600 |
| 60 | 1995-03-02 | 6 | 7 | 720 |
| 70 | 1995-05-06 | 9 | 7 | 150 |

Select * from Client

| Cust_ID | Name | City | IndustryType |
|---|---|---|---|
| 4 | Samsung | Delhi | J |
| 6 | Panasonic | Orange | J |
| 7 | Nokia | Jamshedpur | B |
| 9 | Apple | Jamshedpur | B |

Select * from Salesman

| salesman_id | Name | Age | Salary |
|---|---|---|---|
| 1 | Amir | 61 | 140000 |
| 2 | Balbir | 34 | 44000 |
| 5 | Chander | 34 | 40000 |
| 7 | Damdar | 41 | 52000 |
| 8 | Kumar | 57 | 115000 |
| 11 | Jaggu | 38 | 38000 |

**Question 1:**
Get the names of all salespersons that have an order with Samsung.

```
Select distinct (s.name) from salesman as s
join Items as o on s.Salesman_id = o.Salesman_id
join Client as c on c.Cust_ID = o.Cust_ID
where c.Name = 'Samsung'
```

**Result:**
**name**
Balbir
Kumar

**Question 2:**
Get the names of all salespersons that do not have any order with Samsung.

```
Select name from salesman where name not in  (
Select distinct (s.name) from salesman as s
join Items as o on s.salesman_id = o.salesman_id
join Client as c on c.Cust_ID = o.Cust_ID
where c.Name = 'Samsung')
```

**Result:**
**name**
Amir
Chander
Damdar
Jaggu

**Question 3:**
Get the names of salespersons that have 2 or more orders.

```
Select s.name from salesman as s join Items as o on s.salesman_id = o.salesman_id
group by s.name
having count (*) >= 2
```

**Result:**
**name**
Balbir
Damdar

**Question 4:**
Find the third highest salary:

```
select min (salary)
from (select distinct top 3 salary from salesman
order by Salary desc) as a
```

**Result:**
**(No column name)**
52000

**Question 5:**
Find the third lowest salary:

```
select max (salary)
from (select distinct top 3 salary from salesman
order by Salary asc) as a
```

**Result:**
**(No column name)**
44000

**Example 3:**

Below example has been taken from Stackoverflow:

```
CREATE TABLE A
(
A int,
);

CREATE TABLE B
(
B int,
);
```

```
Insert into A values (1)
Insert into A values (2)
Insert into A values (3)
Insert into A values (4)

Insert into B values (3)
Insert into B values (4)
Insert into B values (5)
Insert into B values (6)
```

Select * from A

| A |
|---|
| 1 |
| 2 |
| 3 |
| 4 |

Select * from B

| B |
|---|
| 3 |
| 4 |
| 5 |
| 6 |

**Question 1:**
What will be the query and result of inner join between tables A and B?

```
Select * from a
INNER JOIN b
on a.a = b.b;
```

| A | B |
|---|---|
| 3 | 3 |
| 4 | 4 |

**Question 2:**
What will be the query and result of full outer join between tables A and B?

```
Select * from a
FULL OUTER JOIN b
on a.a = b.b;
```

| A | B |
|---|---|
| 1 | NULL |
| 2 | NULL |
| 3 | 3 |
| 4 | 4 |
| NULL | 5 |
| NULL | 6 |

**Question 3:**
What will be the query and result of left outer join between tables A and B?

```
Select * from a
LEFT OUTER JOIN b
on a.a = b.b;
```

| A | B |
|---|---|
| 1 | NULL |
| 2 | NULL |
| 3 | 3 |
| 4 | 4 |

Newer Post Home Older Post