SQL Exception Handling

CREATE PROCEDURE dbo.ErrorHandlingTemplate

AS

BEGIN

BEGIN TRY

SET NOCOUNT ON

SET XACT\_ABORT ON

-- Code Which Doesn't Require Transaction

BEGIN TRANSACTION

-- Code which Requires Transaction

COMMIT TRANSACTION

END TRY

BEGIN CATCH

IF @@TRANCOUNT > 0 AND XACT\_STATE() <> 0

ROLLBACK TRAN

-- Do the Necessary Error logging if required

-- Take Corrective Action if Required

THROW --RETHROW the ERROR

END CATCH

END

===============================================================

Union joins the result of two or more tables

It could be on same table or different table but selected column should exist in both tables

UNION performs a DISTINCT on the result set, eliminating any duplicate rows.

UNION ALL does not remove duplicates, and it therefore faster than UNION.

UNION, INTERSECT or EXCEPT operator must have an equal number of expressions

Rules -

SELECT statement with the UNOIN must have the same number of columns.

The columns must also have similar data types.

the columns in each SELECT statement must be in the same order.

NULL is not a data type - this means it is not recognized as an "int", "date" or any other defined data type.

Arithmetic operations involving NULL always return NULL for example, 69 + NULL = NULL.

All aggregate functions affect only rows that do not have NULL values.

==================================================================================

Keys

A primary key cannot be NULL

A primary key value must be unique

The primary key values cannot be changed

The primary key must be given a value when a new record is inserted.

A composite key is a primary key composed of multiple columns used to identify a record unique

Foreign Key references the primary key of another Table

A foreign key can have a different name from its primary key

It ensures rows in one table have corresponding rows in another

Unlike the Primary key, they do not have to be unique. Most often they aren't

Foreign keys can be null even though primary keys can not

SQL Queries

Get nth salary

SELECT \* FROM

(SELECT e.\*, ROW\_NUMBER() OVER (ORDER BY salary DESC) rn FROM Employee e)

WHERE rn = no;

SELECT TOP 1 salary FROM

(SELECT DISTINCT TOP 3 salary FROM #Employee ORDER BY salary DESC) AS temp

ORDER BY salary

SELECT name, salary

FROM Employee e1

WHERE N-1 = (SELECT COUNT(DISTINCT salary) FROM Employee e2

WHERE e2.salary > e1.salary)

Combine result of two or more select table

It could be on same table or different table but selected column should exist in both tables

UNION removes duplicate records abut union UNION ALL does not.

UNION is equal to running distinct on output of UNION ALL

UNION ALL runs faster that UNION because

UNION, INTERSECT or EXCEPT operator must have an equal number of expressions

SELECT statement with the UNOIN must have the same number of columns.

The columns must also have similar data types.

the columns in each SELECT statement must be in the same order.

Find Employees Which are also manager

SELECT e.name, m.name FROM Employee e, Employee m WHERE e.mgr\_id = m.emp\_id;

Write SQL Query to find duplicate rows in a database? and then write SQL query to delete them?

SELECT \* FROM emp a WHERE rowid = (SELECT MAX(rowid) FROM EMP b WHERE a.empno=b.empno)

DELETE FROM emp a WHERE rowid != (SELECT MAX(rowid) FROM emp b WHERE a.empno=b.empno);

Select Key column --> Group by Key column --> Having count(\*) > 1

DELETE E

FROM Employee E

INNER JOIN (Select RowNumber () over (Partitioned by ID group by ID) R

From Employee) Emp

ON E.ID=Emp.ID

Where Emp.R>1

What are window function?

Difference between rank() and dense\_rank() in SQL?

Difference between where and having clause in SQL?

Difference between correlated vs non-correlated subquery?

create table employee(id int, name varchar(50), department varchar(50), manager int, doj date);

insert into employee values(1,'John','IT',9,'05-08-2010');

insert into employee values(2,'Alex','Corp',0,'06-03-2008');

insert into employee values(3,'Linda','IT',9,'07-02-2010');

insert into employee values(4,'Rahul','Purchase',8,'08-12-2010');

insert into employee values(5,'Ismail','Purchase',8,'09-08-2012');

insert into employee values(6,'Zheng','Sales',7,'10-05-2012');

insert into employee values(7,'Reiki','Sales',2,'11-02-2009');

insert into employee values(8,'Aris','Sales',2,'12-08-2011');

insert into employee values(9,'Jena','IT',2,'01-01-2008');

insert into employee values(10,'Bonny','IT',9,'01-01-2008');

select \* from employee;

select id,manager,to\_char(doj,'dd-mon-yyyy') from employee;

1. list the employees who are not managers

select name from employee

where id not in (select manager from employee);

2. manager with only one reportee

select mgr from (select e1.id as mgr ,e2.id as id from employee e1

inner join employee e2 on

e1.id=e2.manager)abc

group by mgr having count(id)=1

3. what is the month with most hiring?

select dt from (select dt,rank() over(order by cnt desc) as rnk from (select to\_char(doj,'mm') as dt ,count(\*) as cnt from employee

group by to\_char(doj,'mm'))) where rnk=1;

4.what is the experience gap between the first employee and the latest?

select max(doj)-min(doj) from employee;

5. name the manager with most reportees?

select mgr from (select mgr,rank() over( order by cnt desc) as rnk from (select mgr,count(id) as cnt from (select e1.id as mgr,e2.id as id from employee e1

inner join employee e2

on e1.id=e2.manager)

group by mgr)) where rnk=1

6.list managers who joined after the reportees

select mgr from (select e1.id as mgr,e2.id as id,e1.doj as mdoj, e2.doj as edoj from employee e1

join employee e2 on

e1.id=e2.manager)

group by mgr

having max(mdoj)>min(edoj)

7.department with most managers and how many?

select department,cnt from (select department,rank() over(order by cnt desc) as rnk,cnt from (select count(distinct manager) as cnt ,department from employee

group by department)) where rnk=1