use pubs

go

select au\_id as 'Author Id', title as 'Book name' from titleauthor join titles on titleauthor.title\_id = titles.title\_id;

select pub\_name as Publisher\_Name, title, ord\_date from publishers JOIN titles on publishers.pub\_id = titles.pub\_id JOIN sales ON titles.title\_id = sales.title\_id;

select pub\_name as Publisher\_Name, title, ord\_date from publishers JOIN titles on publishers.pub\_id = titles.pub\_id JOIN sales ON titles.title\_id = sales.title\_id order by ord\_date;

select p.pub\_name as Publisher\_Name, min(t.pubdate) as First\_published\_Date from publishers p join titles t on p.pub\_id = t.pub\_id group by p.pub\_name ;

select pub\_name, min(ord\_date) as first\_order\_date from publishers p left outer join titles t on p.pub\_id = t.pub\_id left outer join sales s on t.title\_id = s.title\_id group by pub\_name order by min(ord\_date);

select title as Book\_Name, stor\_address as Store\_Address from titles join sales ON titles.title\_id = sales.title\_id join stores on stores.stor\_id = sales.stor\_id order by Book\_Name;

create procedure proc\_firstprocedure

as begin

print 'Hello World!'

end

Go

exec proc\_firstprocedure

create table Products

(id int identity(1,1) constraint pk\_productId primary key,

name nvarchar(100) not null,

details nvarchar(max))

Go

create proc proc\_InsertProduct(@pname nvarchar(100),@pdetails nvarchar(max))

as

begin

insert into Products(name,details) values(@pname,@pdetails)

end

go

proc\_InsertProduct 'Laptop','{"brand":"Dell","spec":{"ram":"16GB","cpu":"i5"}}'

go

select \* from Products

create or alter proc proc\_InsertProduct(@pname nvarchar(100),@pdetails nvarchar(max))

as

begin

insert into Products(name,details) values(@pname,@pdetails)

end

select JSON\_QUERY(details, '$.spec') Product\_Specification from products

create proc proc\_UpdateProductSpec(@pid int,@newvalue varchar(20))

as

begin

update products set details = JSON\_MODIFY(details, '$.spec.ram',@newvalue) where id = @pid

end

proc\_UpdateProductSpec 1, '24GB'

select id, name, JSON\_VALUE(details, '$.brand') Brand\_Name

from Products

create table Posts

(id int primary key,

title nvarchar(100),

user\_id int,

body nvarchar(max))

Go

select \* from Posts

go

create proc proc\_BulInsertPosts(@jsondata nvarchar(max))

as

begin

insert into Posts(user\_id,id,title,body)

select userId,id,title,body from openjson(@jsondata)

with (userId int,id int, title varchar(100), body varchar(max))

end

go

delete from Posts

proc\_BulInsertPosts '

[

{

"userId": 1,

"id": 1,

"title": "sunt aut facere repellat provident occaecati excepturi optio reprehenderit",

"body": "quia et suscipit\nsuscipit recusandae consequuntur expedita et cum\nreprehenderit molestiae ut ut quas totam\nnostrum rerum est autem sunt rem eveniet architecto"

},

{

"userId": 1,

"id": 2,

"title": "qui est esse",

"body": "est rerum tempore vitae\nsequi sint nihil reprehenderit dolor beatae ea dolores neque\nfugiat blanditiis voluptate porro vel nihil molestiae ut reiciendis\nqui aperiam non debitis possimus qui neque nisi nulla"

}]'

select \* from products where

try\_cast(json\_value(details,'$.spec.cpu') as nvarchar(20)) ='i7'

**Afternoon Session:**

pgexercise

1. Select \* from cd.facilities;
2. select name, membercost from cd.facilities;
3. select \* from cd.facilities where membercost > 0;
4. select facid,name,membercost,monthlymaintenance from cd.facilities where membercost <> 0 and membercost < monthlymaintenance/50;
5. Select \* from cd.facilities where name like '%Tennis%';
6. Select \* from cd.facilities where facid in (1,5);
7. select name, case when (monthlymaintenance > 100) then 'expensive' else 'cheap' end as cost from cd.facilities;
8. select memid, surname, firstname, joindate from cd.members where joindate >= '2012-09-01';
9. SELECT distinct surname from cd.members order by surname limit 10;
10. select surname from cd.members union select name from cd.facilities;
11. select joindate from cd.members order by joindate desc limit 1;