



دانشگاه صنعتی امیرکبیر
(پلی تکنیک تهران)

دانشکده مهندسی کامپیوتر و فناوری اطلاعات

دستورکار آزمایشگاه پایگاه داده

طرح: سوگند صدر حقیقی

زیرنظر: دکتر حمیدرضا شهریاری

گزارش زیر برای آموزش دو پایگاه داده‌ی Oracle و SQL Server تهیه شده و دارای سه بخش است. در بخش اول، خلاصه‌ی مطالب ارائه شده در هر جلسه به همراه دستور کار هر جلسه عنوان شده‌اند. دستور کار هر جلسه شامل تمارینی است که به صورت گروه‌های دو نفره سر کلاس انجام می‌گیرد. هر گروه موظف است، گزارش کار هر هفته را که شامل پاسخ به سوالات دستور کار هر جلسه است، تا پیش از شروع جلسه‌ی بعد ارسال کند.

بخش دوم شامل تمارین و پروژه‌های درس است. یک تمرین از قسمت SQL Server و تمرین دیگر از بخش Oracle طراحی شده‌اند. پروژه‌ی پایانی نیز باید با استفاده از Oracle صورت گیرد.

بخش سوم نیز شامل راهنمای گام به گام نصب پایگاه داده‌ی Oracle است.

آزمایشگاه دیتابیس

جلسه‌ی اول

دوره‌ی مباحث درس پایگاه داده – مقدمات ایجاد پایگاه داده و جداول با استفاده از
SQL Server

طرح درس

شماره‌ی جلسه	مباحث مطرح شده در سر کلاس
۱	یادآوری ERD و تبدیل ERD به روابط - کشیدن جداول در sql server، تعریف کلید اصلی، کلید خارجی و constraint در SQL Server
۲	ایجاد view و نحوه زدن query در SQL Server
۳	دستورات در زبان SQL (transact SQL) (دستورات شرطی، حلقه، تعریف متغیر عملیات‌های join، delete و تعریف output) در SQL Server
۴	SQL server function، Stored procedure
۵	Triggerها در SQL Server
۶	دوره‌ی کلی SQL Server (پروژه‌ی ساده از کلیه‌ی قسمت‌ها)
۷	مدتمات oracle (تعریف جداول، تعریف کاربر، تعریف نقش‌ها)
۸	دادن اختیارات به نقش‌های مختلف، کار کردن با SQL Plus برای تعاملات بین کاربر و oracle، نحوه رمزگاری ستون (ایجاد wallet)
۹	تعریف procedure و نحوه auditing در oracle
۱۰	ارائه‌ی دانشجویان به صورت اختیاری (نمره‌ی اضافه)

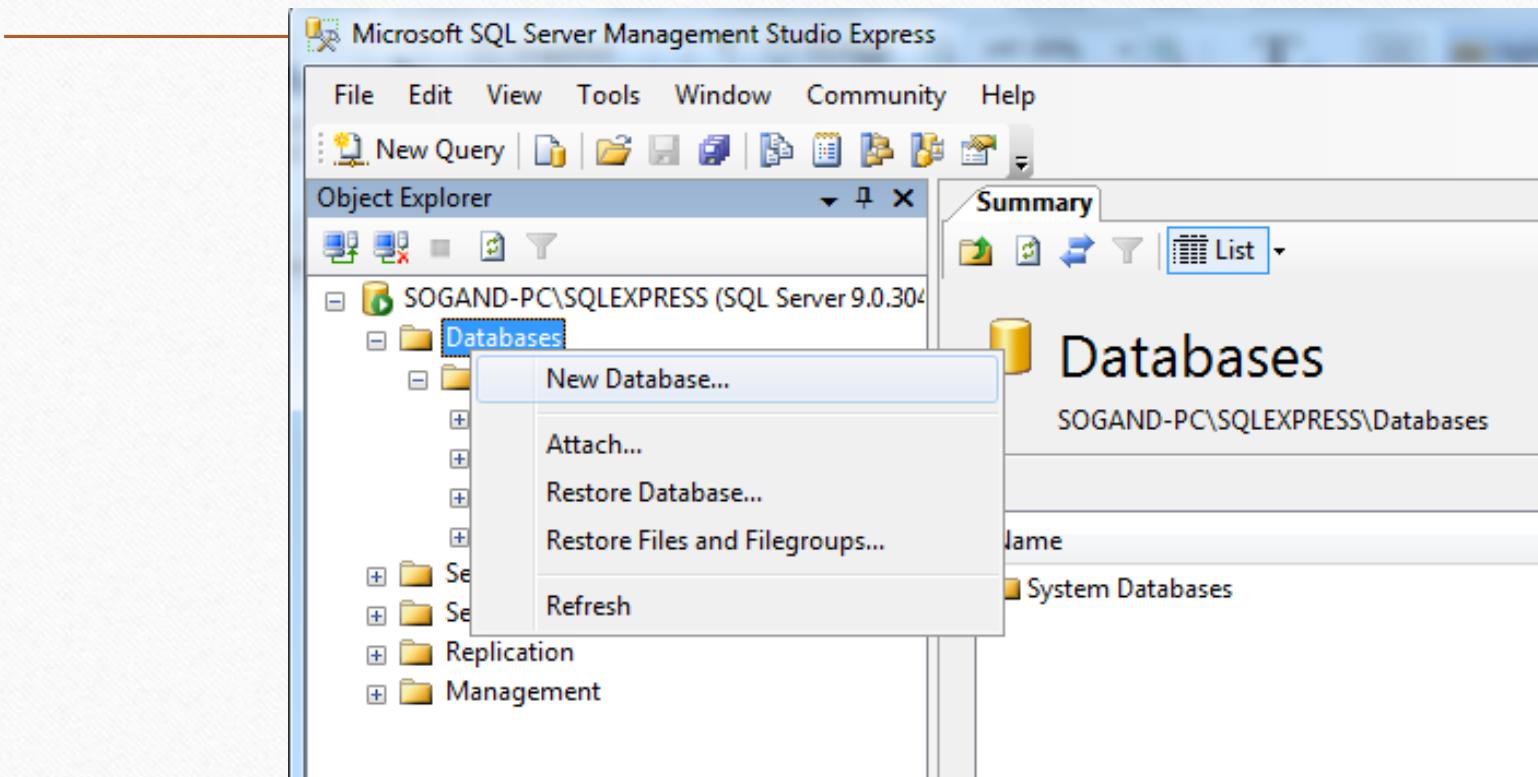
نحوه‌ی نمره دهی

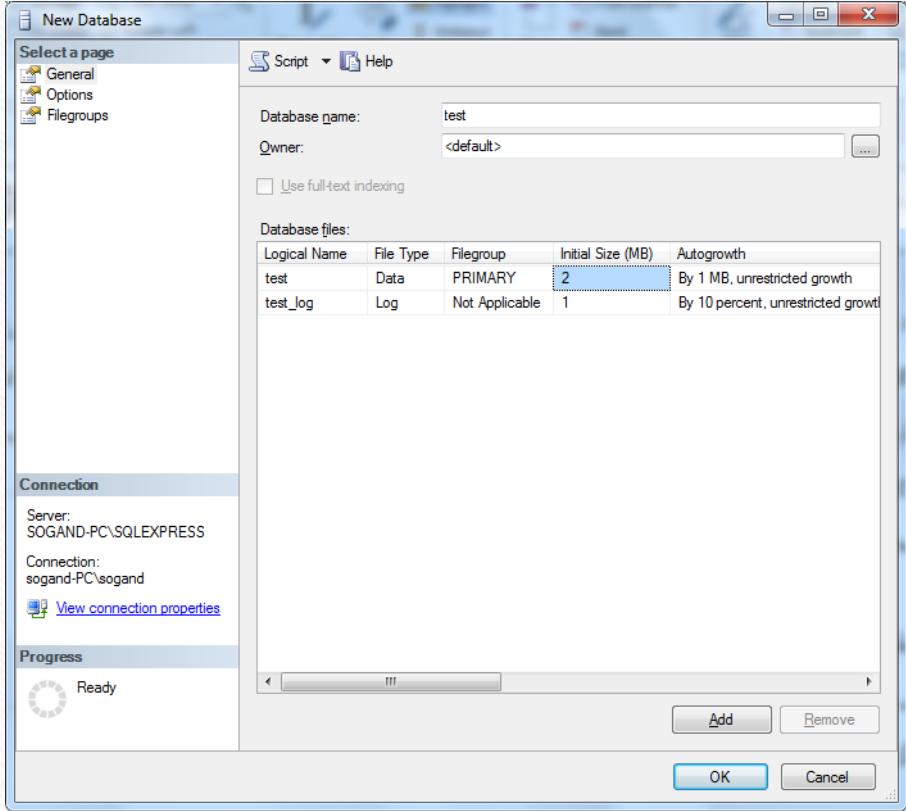
- کار کلاسی(هر جلسه): ۰.۷.
- گزارش کار(هر جلسه): ۰.۵.
- پروژه‌ی ۱(SQL Server): ۳ نمره
- پروژه‌ی ۲(oracle): ۵ نمره

تبديل ER به شمای رابطه ای

- Relationship set:
 - Many-to-Many : Relation(primary keys of entity sets+صفات خودش)
 - Many-to-one : Relation(add the primary key of one side to the relation of many side)
 - one-to-one : Relation(add the primary key of one of them to the relation of the other)

ساخت دیتابیس در SQL Server





- با وارد کردن نام دیتابیس، DBMS دو فایل مختص پایگاه داده شما می سازد.
- فایل Test_Data.MDF محل ذخیره سازی داده های پایگاه
- فایل Test_Log.LDF ثبت تراکنش های پایگاه داده

ساخت جداول

The screenshot shows the Microsoft SQL Server Management Studio Express interface. On the left, the Object Explorer pane displays the database structure of 'SOGAND-PC\SQLEXPRESS (SQL Server 9.0.304)'. In the center, the 'Tables' section shows a table named 'dbo.book' with three columns: 'book_id' (smallint), 'book_name' (nchar(10)), and 'QTY' (smallint). The 'Properties' pane on the right provides details about the table, including its name 'book', database 'test', schema 'dbo', and server 'sogand-pc\sqlexpress'. The 'Identity' section indicates it is an identity column.

Microsoft SQL Server Management Studio Express

File Edit View Tools Window Community Help

New Query | Databases | Tables | Scripts | Reports | Object Explorer

SOGAND-PC\SQLEXPRESS (SQL Server 9.0.304)

Databases

System Databases

- master
- model
- msdb
- tempdb

test

Database Diagrams

Tables

New Table...

Filter

Refresh

Summary

Tables

SOGAND-PC\SQLEXPRESS\

Table - dbo.book Summary

Name	Data Type	Allow Nulls
book_id	smallint	<input type="checkbox"/>
book_name	nchar(10)	<input checked="" type="checkbox"/>
QTY	smallint	<input checked="" type="checkbox"/>

Properties

[Tbl] dbo.book

(Name) book

Database Name test

Description

Schema dbo

Server Name sogand-pc\sqlexpress

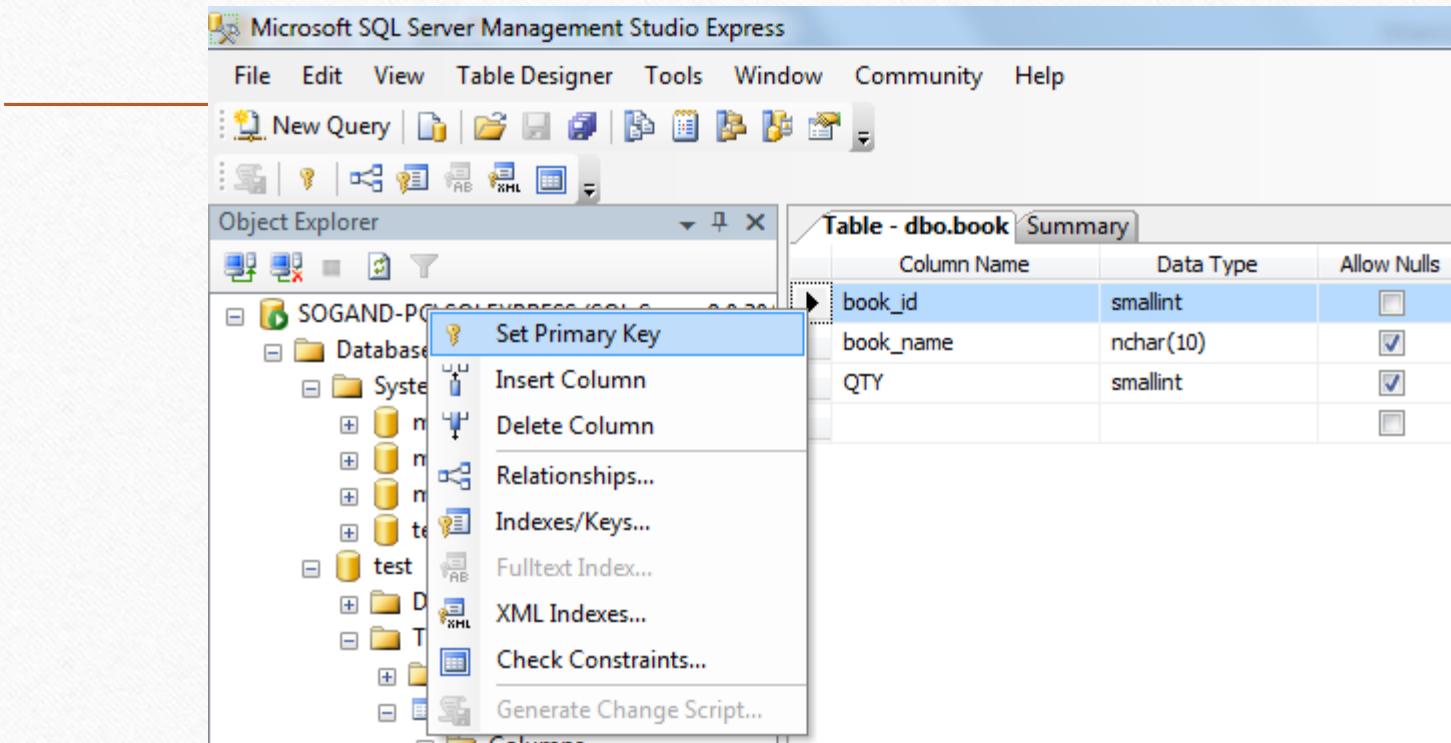
Table Designer

Identity Column

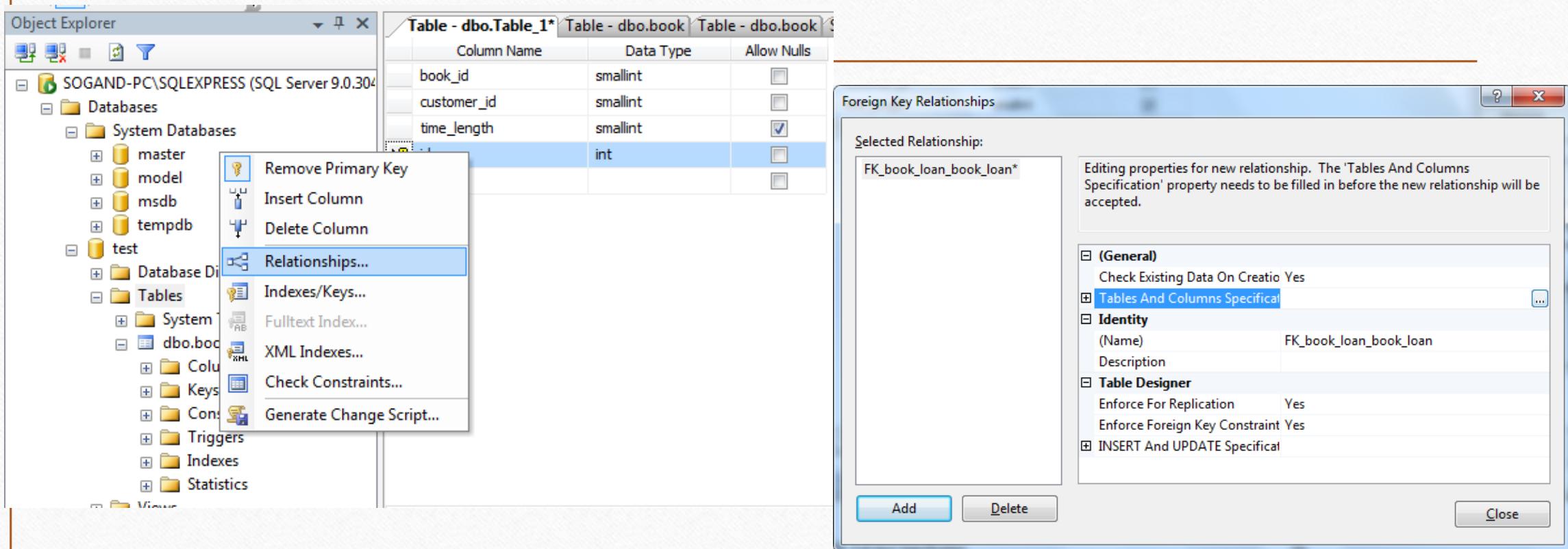
Indexable Yes

(Name)

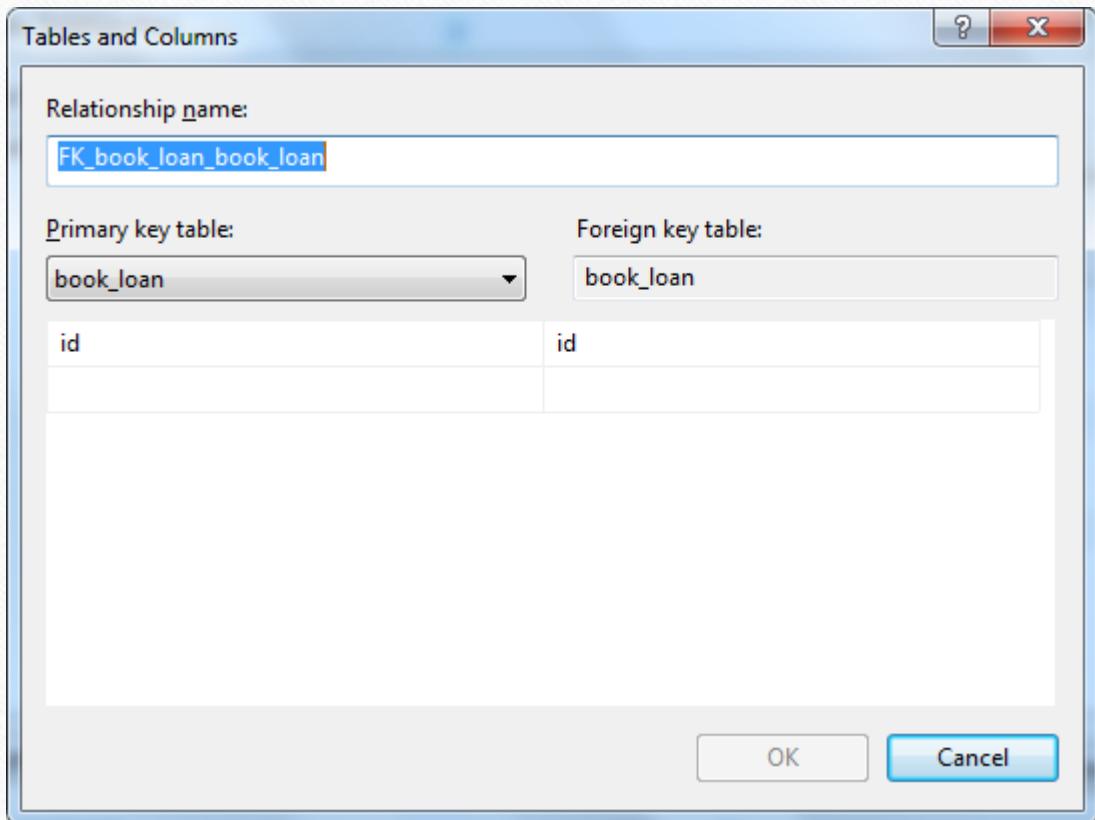
تعریف کلید اصلی و کلید خارجی



تعریف کلید خارجی



تعريف کلید خارجی



The image shows two identical instances of Microsoft SQL Server Management Studio Express running side-by-side. Both windows display the 'Object Explorer' on the left and the 'Table - dbo.book' properties on the right.

Object Explorer:

- SOGAND-PC\SQLEXPRESS (SQL Server 9.0.304)
- Databases
 - System Databases: master, model, msdb, tempdb
 - test
 - Database Diagrams
 - Tables
 - System Tables
 - dbo.book
 - Columns
 - Keys
 - Constraints
 - Triggers
 - Indexes
 - Statistics
 - Views
 - Synonyms
 - Programmability
 - Security
 - Security
 - Server Objects
 - Replication
 - Management

Table - dbo.book Summary:

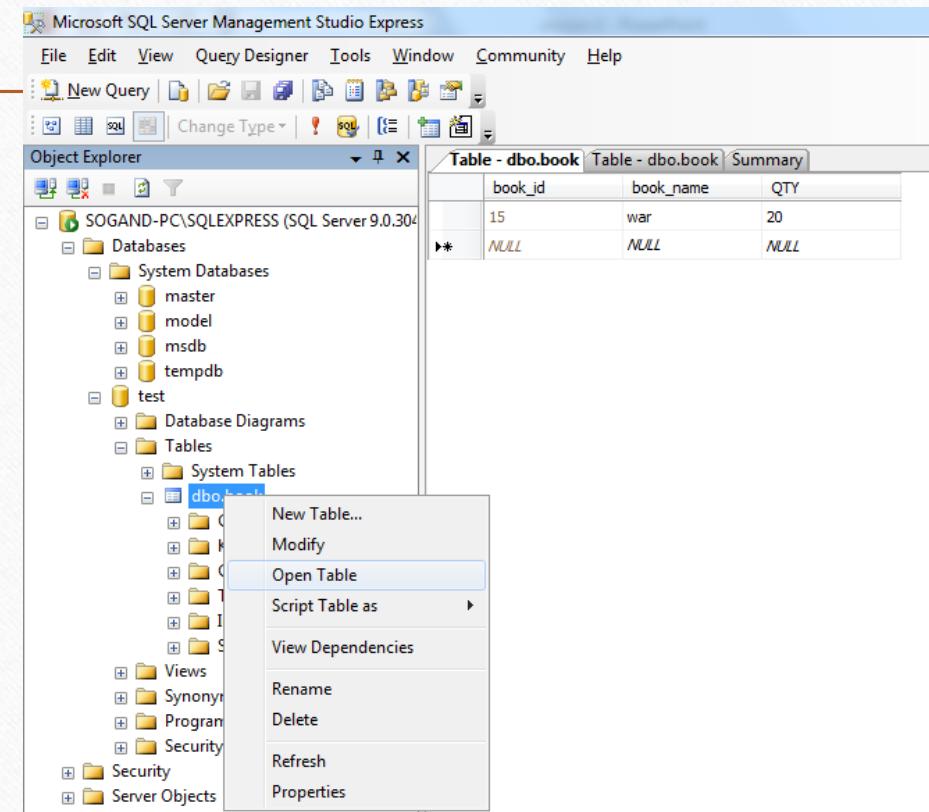
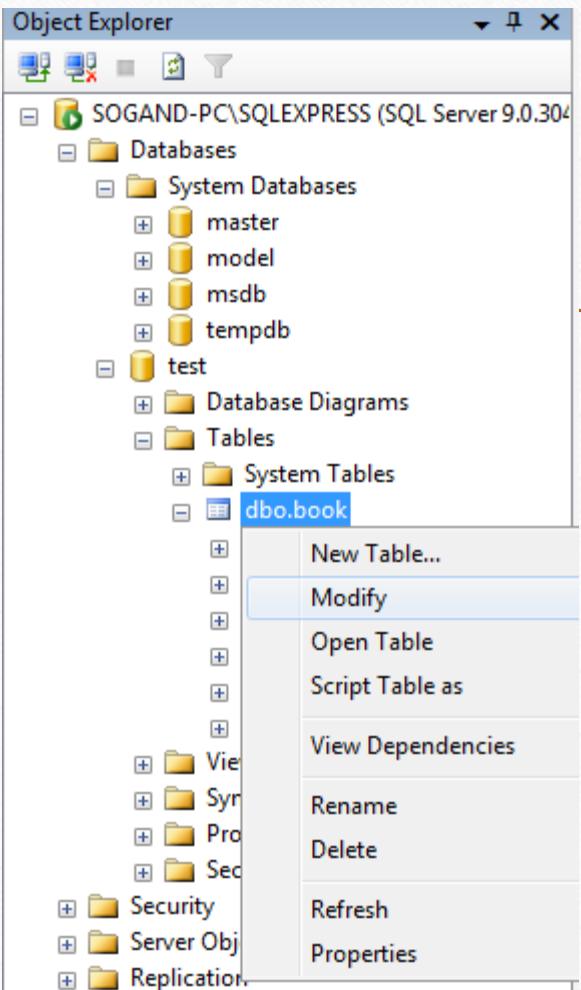
Column Name	Data Type	Allow Nulls
book_id	smallint	<input type="checkbox"/>
book_name	nchar(10)	<input checked="" type="checkbox"/>
QTY	smallint	<input checked="" type="checkbox"/>

Column Properties:

Property	Value
Full-text Specification	No
Has Non-SQL Server Subscriber	No
Identity Specification	No
(Is Identity)	No
Identity Increment	
Identity Seed	
Indexable	Yes
Merge-published	No
Not For Replication	No
Replicated	No

The 'Identity Specification' row is highlighted in blue in both windows.

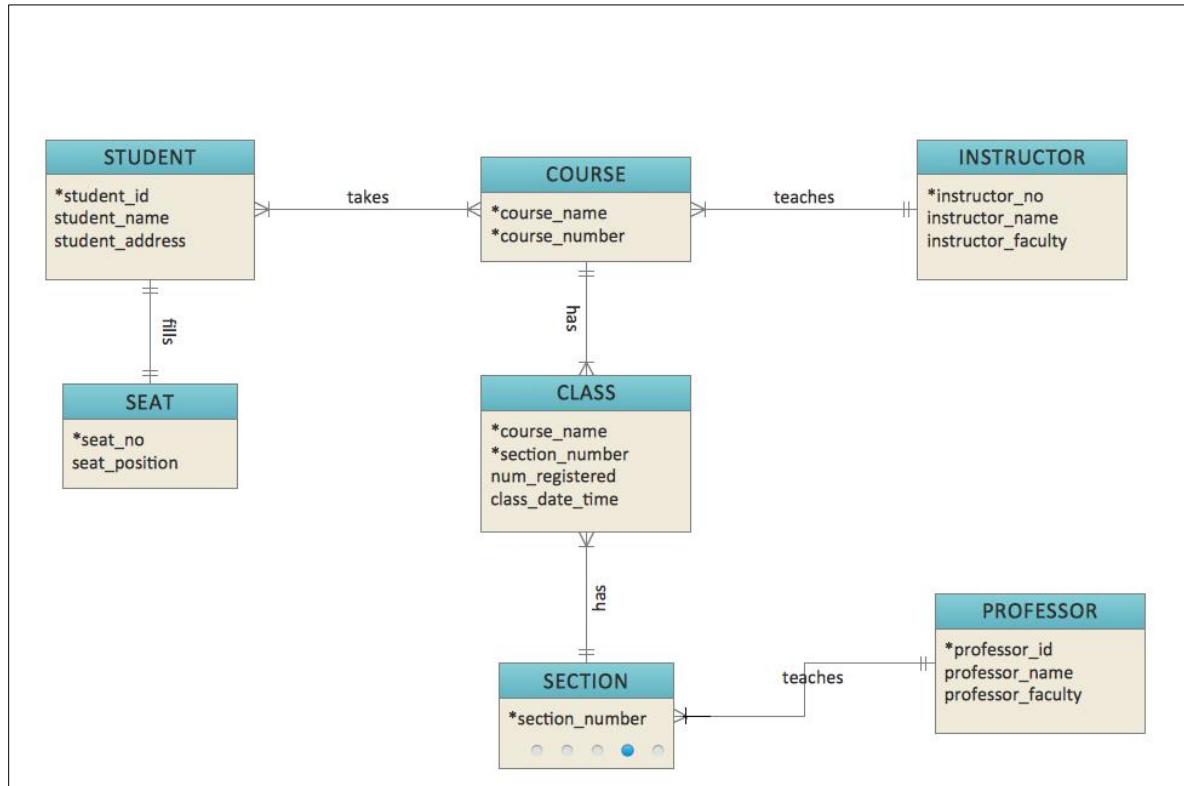
مقداردهی به جداول



فضای لازم	محدوده قابل قبول	نوع داده
۸ بایت	از منفی دو به توان ۶۳ تا دو به توان ۶۳ منهای یک	bigint
۴ بایت	از منفی دو به توان ۳۱ تا دو به توان ۳۱ منهای یک	int
۲ بایت	از -۳۲۷۶۸ تا ۳۲۷۶۷	smallint
۱ بایت	از صفر تا ۲۵۵	tinyint

دستور کار جلسه‌ی اول

- ۱- نمادار ERD زیرا را به مدل رابطه‌ای تبدیل کرده و در SQL Server وارد کنید.
- ۲- برای هر جدول دو داده‌ی نمونه را وارد پایگاه داده کنید.

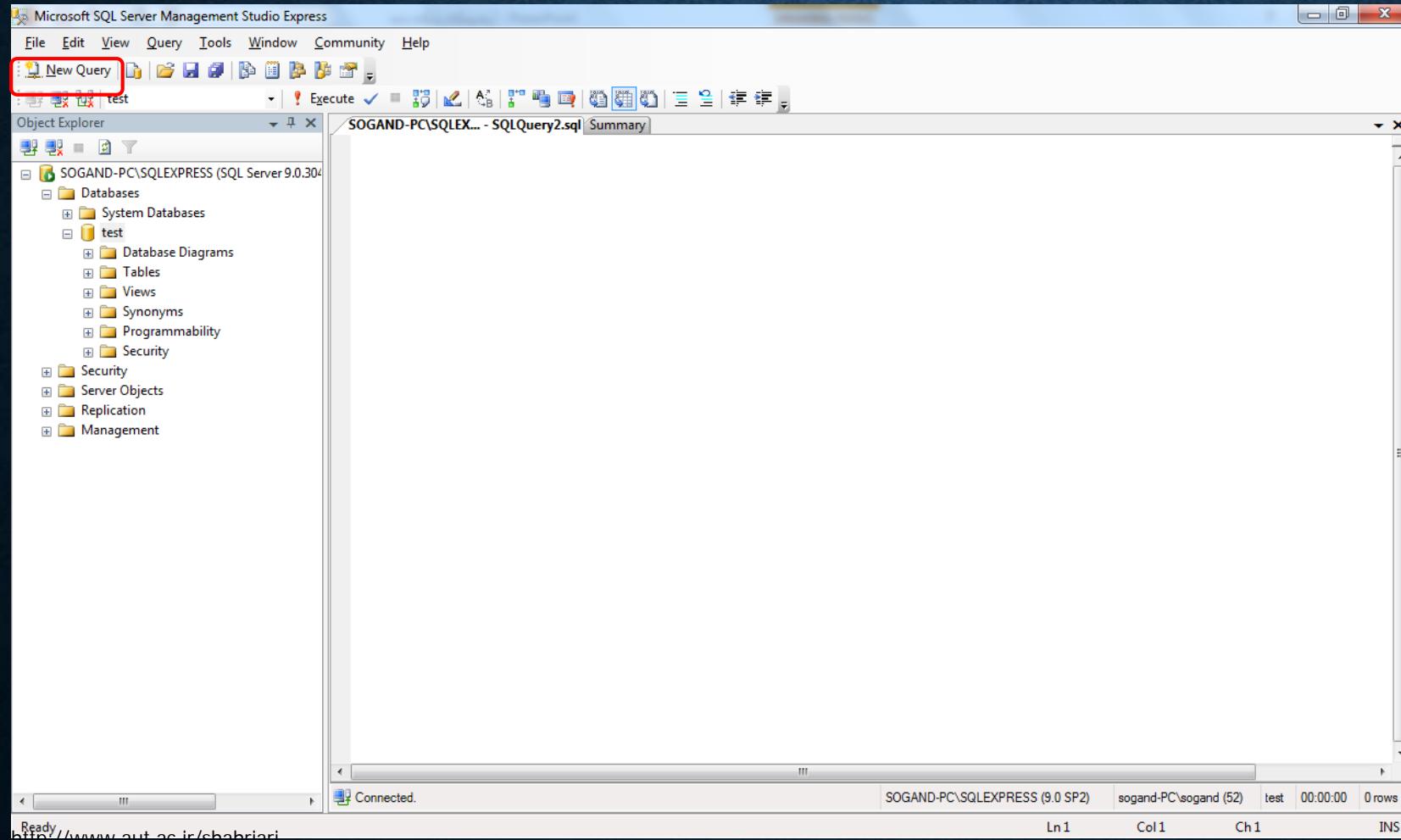


آزمایشگاه پایگاه داده

جلسه ی دوم

انجام پرس جوهای ساده

انجام پرس و جوها



```
Create Database database_name;
```

- ایجاد دیتابیس

```
Drop Database database_name;
```

- حذف دیتابیس

- ایجاد جداول

```
Create Table table_name (
```

```
    Column1 DataType [Nullability] [CONSTRAINT constraint_name CHECK (check_syntax) ],
```

```
    Column2 DataType Default default_value Nullability,
```

```
    ...
```

```
    ColumnN DataType [Nullability],
```

```
    Primary key (one or more columns),
```

```
    foreign key (Column_name) references master_table_name(master_column_name) [ on  
Delete {cascade|..} ] [on Update {cascade|...}]
```

```
);
```

مثال

```
create table sailor(
sailor_name int identity(200,1) not null primary key,
sailor_rank int default 5 not null
)
```

- اگر بیش از یک کلید اصلی موجود است:

```
create table boat(
boat_name varchar(20),
boat_color varchar(30),
primary key(boat_name, boat_color)
)
```

مثال

```
drop table boat;
create table boat(
boat_name varchar(20) primary key, Constraint name
boat_rank int default 60 constraint rank_limit check (boat_rank<70)
)
```

A screenshot of a SQL query window. The query is:

```
insert into boat values ('b',120);
select * from boat
```

The results tab shows the following error message:

Msg 547, Level 16, State 0, Line 1
The INSERT statement conflicted with the CHECK constraint "rank_limit". The conflict occurred in database "test", table "dbo.boat", column 'boat_rank'.
The statement has been terminated.

```
create table reserve(
boat_name varchar(20),
weekday  varchar(20),
foreign key (boat_name) references boat(boat_name)
)
```

تغییر تعریف جدول

- اضافه کردن یک فیلد

Alter table table_name

Add new_column DataType Nullability;

```
alter table reserve  
add sailor_name int not null;
```

```
alter table reserve  
drop column sailor_name;
```

- تغییر نوع فیلد

Alter table table_name

Alter Column column_name New_DataType [Nullability]

- اضافه کردن محدودیت

Alter table table_name

Add CONSTRAINT constraint_name CHECK (check_syntax)

- حذف محدودیت:

Alter table table_name

Drop Constraint constraint_name

```
create view v1 as  
select * from boat;
```

- ساختن **view** باید اولین **statement** در یک **query batch** باشد.

دستور کار جلسه‌ی دوم

۱- جداول زیر را ایجاد کنید:

Sailor	<u>Sailor_name</u>	Sailor_rank	
Boat	<u>Boat_name</u>	Boat_color	Boat_rank
Reserve	<u>Sailor_name</u>	<u>Boat_name</u>	Weekday

• Sailor_name کلید اصلی جدول Sailor است.
 • Boat_name کلید اصلی جدول Boat است.
 • با هم Sailor_name و Boat_name هستند و به ترتیب کلید خارجی به جداول Reserve و Sailor هستند.
 • "فیلد رتبه" برای قایق‌ها نشان دهنده حداقل رتبه‌ای است که یک ملوان برای رزرو آن قایق نیاز دارد.

الف) محدودیت‌های زیر را اعمال کنید:

Sailor_name از نوع int بوده، identity و از مقدار ۳۰۰ شروع می‌شوند.

Boat_color از نوع varchar و با حداکثر ۳۰ کاراکتر

Boat_rank از دارای مقدار حداقل ۷۰ و حداکثر ۱۵۰ باشند.

ب) برای هر جدول ۴ داده‌ی مختلف را وارد کنید.

ج) پرس و جوهای زیر را بیابید:

۱- نام کلیه‌ی قایق‌هایی که در روز شنبه به آب انداخته شده‌اند.

۲- ایجاد دید(view) که نام کشتی، نام قایقران و رنگ کشتی را نشان دهد.

۳- نمایش رنک کلیه‌ی قایق‌ها.

۴-نمایش نام قایق هایی که در روزهای شنبه یا دوشنبه به آب اندخته شده‌اند.

۵-رنگ تمامی قایق هایی که در روزی که نام آن با 'S' شروع می‌شود، به آب اندخته شده‌اند.

۲-جداول زیر را وارد کنید:

Emp

ستون	نوع	محدودیت
Emp_id	عدد صحیح	Primary Key, Identity
Name	رشته حداکثر ۵۰ کاراکتر	
Salary	عدد صحیح	
DeptNo	عدد صحیح	Foreign Key

Dept

ستون	نوع	محدودیت
DeptNo	عدد صحیح	Primary Key
Name	رشته ۵۰ کاراکتر	

Prj

ستون	نوع	محدودیت
Prj_id	عدد صحیح	Primary Key, Identity
Emp_id		

الف) پروژه های زیر را در جدول وارد کنید:

- (۱) پروژه‌ی A: کارمند^b(1)، کارمند^a(2)، کارمند^c(2)
- (۲) پروژه‌ی B: کارمند^c(3)، کارمند^d(3)، کارمند^a(2)، کارمند^e(2)، کارمند^b(1)
- (۳) پروژه‌ی C: کارمند^a(1)، کارمند^f(2)
- (۴) پروژه‌ی D: کارمند^g(2)، کارمند^h(3)، کارمند^a(4)، کارمند^f(1)، کارمند^k(3)

ب) دستورات زیر را اجرا کنید:

- (۱) لیستی از پروژه هایی که کمتر از ۴ کارمند در آن ها کار می کنند.
- (۲) لیستی از کارمندان به همراه نام بخش مربوطه، نام پروژه ها.
- (۳) مجموع حقوق افرادی که در پروژه‌ی B شرکت دارند.
- (۴) لیستی از پروژه ها به همراه حقوق متous افرادی که در آن شرکت دارند.
- (۵) لیست پروژه هایی که manager در آنها دخیل است.

1

جلمه ی سوم

آزمایشگاه پایگاه داده

انجام پرس جوهای پیچیده تر - دستورات **T-SQL**

دستور DELETE

- این دستور برای حذف یک یا چند ردیف جدول استفاده می شود. شکل کلی آن به صورت زیر است:

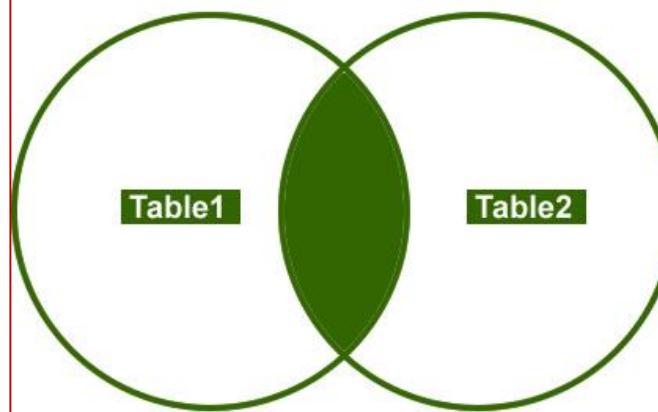
```
DELETE [FROM] table_or_view_name  
WHERE search_condition
```

```
DELETE FROM tblSailor  
WHERE sailor_rank<5
```

- دستور **truncate** برای حذف تمامی ردیف های یک جدول استفاده می شود و نسبت به **Delete** سرعت بیش تری دارد.

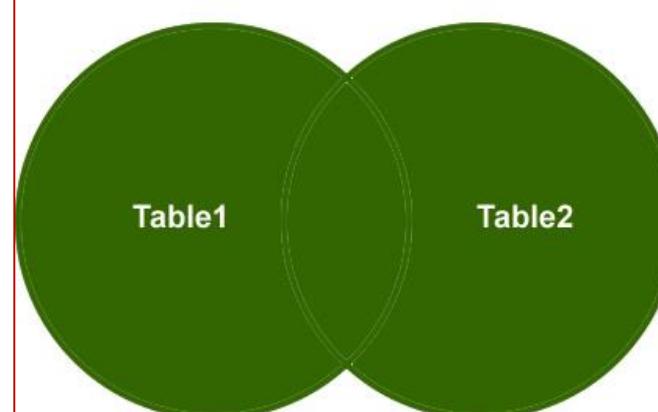
JOIN

INNER JOIN



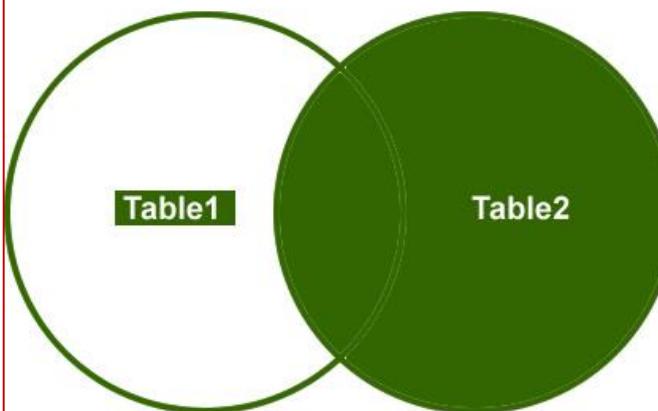
```
SELECT *  
FROM Table1 t1  
INNER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

FULL OUTER JOIN



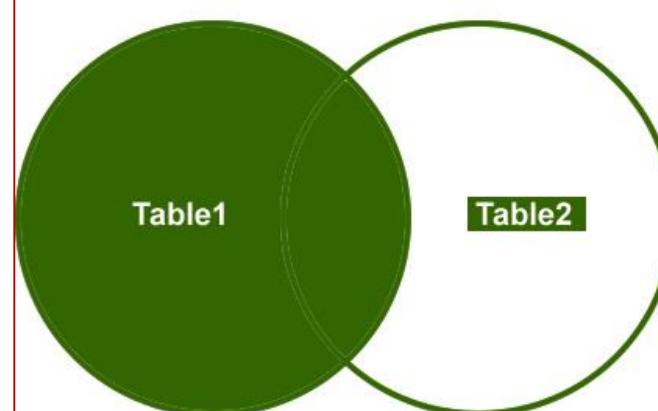
```
SELECT *  
FROM Table1 t1  
FULL OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

RIGHT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
RIGHT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

LEFT OUTER JOIN



```
SELECT *  
FROM Table1 t1  
LEFT OUTER JOIN Table2 t2  
ON t1.Col1 = t2.Col1
```

TRANSACT-SQL

یک زبان پرس و جو استاندارد است که برای واکشی اطلاعات از آن استفاده می شود.

در ای قابلیت های بیشتری نسبت به **SQL** است:

- برنامه نویسی روانه ای
- تعریف متغیر محلی

تعريف متغير

- زبان **T-SQL** امکان تعريف متغيرها برای استفاده در رويه هاي پرس و جو خراهم مي آورد:
- تعريف متغير: يكى از اشيا در **SQL Server** است که برای حمل و نگهداری مقادير مفرد به کار مى روند.
- از علامت **@** به منظور معرفی يك متغير استفاده می شود.
- برای تعريف يك متغير از قالب زير استفاده می شود:

Declare @VariableName

- برای مقدار دهی به متغيرها از يكى از قالب هاي زير استفاده می شود:

Select @VariableName = Value

Set @VariableName = Value

مثال ایجاد از متغیرها

```
declare @student_name varchar(20);
set @student_name='mary';
select @student_name;
```

```
declare @name varchar(30);
select @name= names from session3_DB.dbo.t1
where studentid=8831033;
select @name
```

	Results	Messages
	(No column name)	
1	mary	

	Column Name	Data Type	Allow Nulls
▶	names	varchar(20)	<input checked="" type="checkbox"/>
	id	int	<input type="checkbox"/>
	familyname	varchar(30)	<input type="checkbox"/>

	names	id	familyname	studentid
	maryam	8	ghorbany	8831033
	flor	9	nazary	9091312

	(No column name)
1	maryam

مثال استفاده از متغیرها

```
declare @name varchar(30);
select @name= names from session3_DB.dbo.t1
select @name;
```

	(No column name)
1	flor

زبان "کنترل جریان"

- زبان کنترل جریان بخشی از دستورات **T-SQL**، ا در برابر می‌گیرد که جریان اجرای دستورات را در یک بلوک دستورات، یک **Stored Procedure** و یا یک **User Defined Function** بر عهده می‌گیرد.
- بدون استفاده از دستورات این زبان، دستورات زبان **T-SQL** به صورت ترتیبی و پشت سرهم اجرا می‌شوند. با استفاده از این دستورات، امکان تعریف شرط، حلقه، پرسن از دستورات و نظایر آن فراهم می‌شود. کنترل جریان از دستورات زیر تشکیل شده است:

- **BEGIN...END**
- **WHILE**
- **RETURN**
- **WAITFOR**

- IF...ELSE**
- BREAK**
- GOTO**
- CONTINUE**

IF ... ELSE <ستور>

- برای تعریف شرط به کار می رود .
- قالب کلی دستور به شکل زیر است :

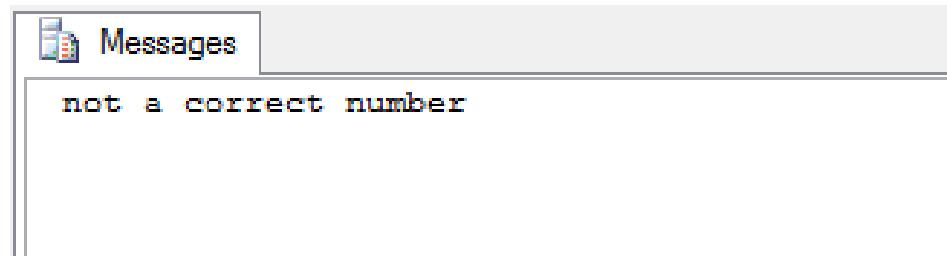
If عبارت شرطی

کد دستورات مورد نظر

[ELSE] [کد دستورات مورد نظر]

مثال:

```
declare @number int;  
set @number=12;  
  
if @number=10  
print '10';  
else print 'not a correct number';
```



```

declare @number int;
set @number=12;

if @number=10
insert into
session3_DB.dbo.tblboat(boatname,boatcolor,boatrank)values('R4','black',13);
else insert into
session3_DB.dbo.tblboat(boatname,boatcolor,boatrank)values('R5','gray',9);

select *from session3_DB.dbo.tblboat;

```

	boatname	boatcolor	boatrank
1	R1	red	6
2	R2	blue	10
3	R3	Green	12
4	R5	gray	9

WHILE *(ستو،)*

While عبارت شرطی هسته، در ماقعه

[Begin]

کد ستوات موردنظر

[End]

مثال:

```
declare @number int;  
set @number=100;  
while @number>95  
    begin  
        print @number;  
        set @number=@number-1;  
    end
```

100
99
98
97
96

CASE (ستور)

مقدار خاصی را بر اساس شرایط تعریف شده، در **ستور** **Select**، باز می‌گرداند.

قالب کلی این **ستور** به اشکال زیر است :

حالت اول :

Case	عبارت		
	When	مقدار اولیه	Then
			مقدار بازگشتی
	...		
	[Else]		
End	[As	[نام	

حالت دو:

Case			
	When	عبارت شرطی	Then
			مقدار بازگشتی
	...		
	[Else]		
End	[As	[نام	

CASE ,ویس ; ا جلیں

boatname	boatcolor	boatrak
R1	red	6
R2	blue	10
R3	Green	12
R5	gray	9

13
select boatcolor,bname=(case boatname
when 'R1' then 'AAAA'
when 'R2' then 'BBBB'
when 'R3' then 'CCCC'
when 'R5' then 'DDDD'
else 'others'
end)
from session3_DB.dbo.tblBoat;

	boatcolor	bname
1	red	AAAA
2	blue	BBBB
3	Green	CCCC
4	gray	DDDD

WAITFOR

- WAITFOR {DELAY 'time' | TIME 'time'}

--Pause for ten seconds

```
WAITFOR delay '000:00:10';
```

```
PRINT 'Done';
```

--Pause until a certain time

```
WAITFOR time '12:00:00';
```

```
PRINT 'It is noon';
```

عبارت output

عبارت **Output** به همراه دستورات تغییر اطلاعات به کار می رود:

Delete,update,insert

15

```
SQL Statement Syntax
INSERT|DELETE|UPDATE ...
OUTPUT {DELETED | INSERTED | from_table_name} . { * | column_name } [,...n ]
INTO { @table_variable | output_table } [WHERE ...]
```

دو جدول **deleted** و **inserted** وجود دارد که دارای ساختار مشابه جدول اصلی هستند و می توان تمام و یا بعضی از خیلدهای آن را انتخاب کرد. جدول **deleted** کاوی مقادیر جدید و به روز شده است و جدول **inserted** مقادیر قبلی را دارد.

مثال استفاده از OUTPUT

	boatname	boatcolor	boatrank
	R1	red	4
	R2	blue	8
	R3	Green	12

▪ می فواهیم به قایق هایی که، تبهی کم تر از ۱۰ دارند، دو، تبه اضافه کنیم و سپس لیستی از، تبه های فعلی و قبلی، کوردهای تغییر کرده، را برگردانیم:

```
declare @temp table(  
    nameb varchar(20),  
    color varchar(20),  
    old_rank int,  
    new_rank int);  
  
update session3_DB.dbo.tblBoat  
set boatrank=boatrank+2  
output inserted.boatname,inserted.boatcolor,deleted.boatrank,inserted.boatrank  
into @temp  
where boatrank<10;  
select *from @temp;
```

	nameb	color	old_rank	new_rank
1	R1	red	4	6
2	R2	blue	8	10

	names	id	familyname	studentid
1	maryam	8	ghorbany	8831033
2	flor	9	nazary	9091312
3	ali	10	ghorbanzadeh	8831020
4	mohammad	11	mohammadi	92131098

	boatname	boatcolor	boatrakn
1	R1	red	6
2	R12	yellow	19
3	R2	blue	10
4	R3	Green	12
5	R5	gray	9
6	R6	white	6
7	R8	white	6
8	R9	white	16

Begin transaction t1

```
insert into session3_DB.dbo.tblBoat(boatname,boatcolor,bo
values('R12','yellow',19);
```

```
select * from session3_DB.dbo.tl;
```

```
select * from session3_DB.dbo.tblBoat;
```

```
commit transaction t1 ;
```

	boatname	boatcolor	boatrakn
1	R1	red	6
2	R2	blue	10
3	R3	Green	12
4	R5	gray	9
5	R6	white	6
6	R8	white	6
7	R9	white	16

دستور کار جلسه ی سوم

۱- جدول زیر را با استفاده از دستورات sql ایجاد کرده و داده های داده شده را در آن وارد کنید.(P-ID) از نوع auto increment بوده و زوج (نام، نام خانوادگی) برابر کلید اصلی هستند.

P_Id	LastName	FirstName	Address	City
1	Hansen	Ola	Timoteivn 10	Sandnes
2	Svendson	Tove	Borgvn 23	Sandnes
3	Pettersen	Kari	Storgt 20	Stavanger
4	Nilsen	Tom	Vingvn 23	Stavanger

جدول persons

پرس و جوهای زیر را پاسخ دهید:

(الف) افراد را بر حسب نام خانوادگی به ترتیب صعودی مرتب کنید.

(ب) در یک transaction فیلد جدید شماره تلفن را از نوع nvarchar اضافه کنید. شماره تلفن ها حتما باید با پیش شماره ۰۰۱ شروع شوند(تعريف قید) و شماره تلفن های دلخواهی را به ۴ داده ای درون جدول نسبت دهید(طول شماره تلفن های اضافه شده برابر باشد).

(پ) با استفاده از دستور case نام، نام خانوادگی و آدرس افراد را نشان دهید به طوری که هر آدرس فعلی(نام خیابان)، نام میدان و شماره ای پلاک را شامل شود(این موارد را به دلخواه وارد کنید).

(ت) داده ای زیر را با شماره تلفن دلخواه وارد جدول کرده و ۳ فیلد اول جدول را به گونه ای نمایش دهید که بر حسب نامشان به صورت صعودی مرتب شده اند.(در یک transaction)

7	Tjessem	Jakob	Nissestien 67	Sandnes
---	---------	-------	---------------	---------

توجه داشته باشید که P-ID داده ای جدید وارد شده برابر ۷ است.

(ث) اشخاصی را که شهری زندگی آن ها با حرف "S" شروع می شود را پس از ده ثانیه مشخص کنید.

(ج) روالی بنویسید که در آن متغیر temp از نوع int آخرین مقدار P_ID در جدول را بگیرد، و به آن تعداد عبارت "okay" چاپ شود.

(چ) داده ای زیر با شماره تلفن آن را در نظر بگیرید، در صورتی که شماره تلفن آن از Tjessem کوچک تر است، - ID آن را ۶ قرار داده و در غیر این صورت P-ID را ۸ قرار دهید.

taylor	Jackson	Nisseisten87	Sandnes
--------	---------	--------------	---------

۲- جدول زیر را ایجاد کرده و داده های زیر را وارد کنید(کلید اصلی=student_id)

name	student_id	grade
R1	8831047	12
	8831043	10
R2	8831031	15
R3	8831051	16
R4	8831012	11
R5		

می خواهیم به کسانی که نمره‌ی کمتر از ۱۵ دارند، ۲ نمره اضافه کنیم، به طوری که نمره‌ی جدید و نمره‌ی قدیمی آن‌ها نیز نمایش داده شود.(راهنمایی: استفاده از output

آزمایشگاه پایگاه داده

جلسه ی چهارم

روالهای ذخیره شده - توابع

روال های ذخیره شده (Stored Procedure)

- قطعه برنامه هایی هستند که عمدتاً از دستوراتی به زبان T-SQL تشکیل می شوند و می توانند هرگاه توسط کاربر فراخوانی می شوند ، عملیات موردنظر وی را بر روی اشیاء مختلف با انک اطلاعاتی انجام دهند .
- مزایای استفاده از روال ها عبارتند از :
 - کد های مربوط به دسترسی به داده ها در یک مکان متمرکز می شوند به جای آن که در بخش های مختلف نرم افزار کاربردی پرآنده باشند و بنابراین تغییر دادن و خطا یابی آن ها ساده تر است.
 - امکان استفاده از کد در بخش های مختلف نرم افزار کاربردی خواهیم شود.
 - روال های ذخیره شده نسبت به دستورات SQL موردنی معمولاً سریع تر و بالاترایی مناسب تری انجام می شوند.
 - بهبود امنیت

تعریف یک روال بدون پارامتر

- شکل کلی تعریف روال ذخیره شده به صورت زیر است:

```
CREATE PROCEDURE procedure_name  
AS <sql-statement(s)>
```

```
create procedure firstSp  
as select * from tbl_sailor  
where sailor_rank>5
```

- مثال:

```
EXEC firstSP
```

- برای اجرای یک استفاده می شود:

تغییر روال های ذخیره شده

- برای تغییر یک روال ذخیره شده می توان از دستور **ALTER PROCEDURE** استفاده کرد. با استفاده از این دستور می توان کلیه ای اجزای یک روال ذخیره شده به جز نام آن را تغییر داد:

```
Alter PROCEDURE procedure_name  
as ...
```

- حذف یک روال ذخیره شده:
- برای حذف یک روال ذخیره شده از دستور **DROP PROCEDURE** استفاده میشود.

تعریف روال پارامتردار

- شکل کلی تعریف یک روال پارامتردار به صورت زیر است:

```
CREATE PROCEDURE procedure_name  
@param_name data_type [=default_value] [OUTPUT] [, ...]  
AS <sql-statement(s)>
```

- برای تعریف هر پارامتر باید نام و نوع آن پارامتر مشخص شود.
- پارامترهایی که می توان تعریف کرد: ۱- ورودی ۲- خروجی (این نوع پارامترها با کلمه **OUTPUT** مشخص می شوند)

روال پارامتردار

- مثال: روال، و به رو پارامتری از نوع رشته ای دریافت کرده و ملوانی با آن نام را حذف می کند:

```
create procedure secondSP  
@temp varchar(20)  
as delete from tbl_sailor  
where sailor_name=@temp
```

```
EXEC secondSP 'ali'
```

```
EXEC secondSP @temp = 'ali'
```

- خرافه ای این روال به صورت زیر است:

نحوه‌ی تعریف روال‌های ذخیره شده با پارامتر خروجی

- در این مثال نحوه‌ی به کارگیری پارامترهای فروجی نشان داده شده است. روال موردنظر تعداد ملوانانی که، نک بیشتر و یا مساوی، نک داده شده دارند، را برمی‌گرداند.

```
create procedure thirdSP  
@inRank int, @countSailor int output  
as  
select @countSailor=count(*)  
from tbl_sailor  
where sailor_rank>=@inRank
```

```
declare @res int  
exec thirdSP  
@inRank=5,  
@countSailor=@res output  
print @res
```

```
declare @res int  
exec thirdSP 5 , @res output  
print @res
```

برای اجرا:

تعریف تابع(function)

- فرمت کلی تعریف تابع به صورت زیر است:

Create Function

نمود نظر

(فهرست پارامترها)

تایپ فروجی

As

Begin

T-SQL

لیست دستورات

End

مثال تابع

```
create function sel (@id int)
returns int
as
begin
    declare @num int;
    select @num= BSD from book where ID=@id
    return @num;
end
```

```
declare @return int;
exec @return=sel @id=2
print @return;
```

id	bookname	BSD
1	war and peace	22
2	general	13
3	red and black	10

13

تفاوت های تابع با روالهای ذخیره شده

- Function must return a value but in Stored Procedure it is optional(Procedure can return zero or n values).
- Functions can have only input parameters for it whereas Procedures can have input/output parameters .
- Functions can be called from Procedure whereas Procedures cannot be called from Function

معرفی برخی توابع

- زبان **T-SQL** از برفی توابع پشتیبانی می کند که کار را، احت تر می کند:
 - توابع تاریخ و زمان:
- برای مقایسه ی دو تاریخ از این توابع استفاده می شود: **DATEDIFF()**

DATEDIFF(datepart,firstdate,seconddate)

که **datepart** معیار، مقایسه ی دو تاریخ است که می تواند **ss**(ثانیه)، **mi**(دقیقه)، **hh**(ساعت)، **dd**(روز)، **mm**(ماه) و **yy**(سال) باشد.

- برگرداندن تاریخ جاری: **GETDATE**

معرفی برخی توابع

توابع رنک دهی (Ranking Function) : توابعی وجود دارند که می توانیم از آنها برای رنک دهی استفاده کنیم. این توابع متفاوت بوده و نحوه رنک دهی آنها متفاوت است.

- **ROW_NUMBER()** : به ردیفهای خروجی به ترتیب رتبه می دهد.
- **RANK()** : این تابع اگرچه به ردیفهای یکسان، رنک یکسان می دهد اما شمارنده در پس زمینه همچنان در حال شمارش است و برای ردیف جدید (با مقدار جدید)، رنکی که حاوی مقدار شمارنده است ثبت می شود.
- **DENSE_RANK()** : مشابه با **RANK()** با این تفاوت که شمارنده درونی در هنگام رنک دهی به ردیفهای یکسان، متوقف می شود.

توابع رنک دهی

- الگوی کلی آن به صورت زیر است:

```
<FUNCTION_NAME>() over (order by <order_by_list>)
```

```
select B.boat_name, R.sailor_name,B.boat_color,  
      ROW_NUMBER() over (order by B.boat_name) as rowNumber,  
      RANK() over (order by B.boat_name) as rankNumber,  
      DENSE_RANK() over (order by B.boat_name) as denseRank  
from tblBoat B,tblReserve R  
where B.boat_name=R.boat_name
```

دستور کار جلسه ی چهارم

در هر قسمت حتما عکس نتیجه ی دریافتی را درج کنید.

تمرین ۱: رویه ای بنویسید که وضعیت نمره ای کلاس را با بررسی جدول tblstudent که شامل فیلدهای نام دانشجو(stuName)، شماره دانشجو(StuID) و نمره ی دانشجو(stuGrade) است به عنوان خروجی برگرداند. اگر تعداد دانشجویانی که نمره ی کمتر از ۱۰ گرفته اند، حداکثر یک نفر باشد وضعیت خوب(GOOD) است. اگر این تعداد دو یا سه نفر باشد وضعیت کلاس نرمال(Normal) است و در نهایت اگر این تعداد بیش تر بود، وضعیت کلاس بد(Bad) است.
توجه: شماره ی دانشجویی کلید اصلی است. و stuGrade از نوع real است.

تمرین ۲. روالی بنویسید که پارامتر num از نوع int را از ورودی بگیرد، با تعداد نمرات زیر ۱۰ از tblStudent مقایسه کند، اگر تعداد نمرات زیر ده از num کمتر بود، نمرات بین ۹ و ۱۰ را یک نمره اضافه کند، در غیر این صورت نمرات بین ۹,۵ تا ۱۰ را ۰,۵ اضافه کند.

تمرین ۳. روالی بنویسید که دو عدد را swap کند. روال را با اعداد ۱۲۳ و ۲۴ امتحان کنید.

تمرین ۴. تابعی بنویسید که نام یک دانشجو را به عنوان ورودی بگیرد و نمره ی آن دانشجوی خاص را برگرداند. دانشجویی با نام ali با شماره ی دانشجویی ۹۰۱۲۳۴۵ و نمره ی ۱۴ را وارد جدول کنید. تابع نوشته شده را با 'ali' امتحان کنید.

تمرین ۵. جدول زیر را وارد کرده و با استفاده از توابع رنکدهی گفته شده، بر اساس postal code رنک دهی کنید.

FirstName	LastName	PostalCode
Michael	Blythe	98027
Linda	Mitchell	98027
Jillian	Carson	98027
Garrett	Vargas	98027
Tsvi	Reiter	98027
Shu	Ito	98055
José	Saraiva	98055
David	Campbell	98055
Tete	Mensa-Annan	98055
Lynn	Tsoflias	98055
Rachel	Valdez	98055
Jae	Pak	98055
Ranjit	Varkey Chudukatil	98055

تمرین ۶. تابعی بنویسید که زمان جاری را به عنوان ورودی بگیرد و قسمت روز ۴ روز بعد را به عنوان خروجی برگرداند.(راهنمایی: استفاده از DATENAME).

تمرین ۷. آیا ها را می توان با دستوراتی مانند insert/update/delete function به کار برد(امتحان کنید و نتیجه را همراه با عکس ذکر کنید).

آزمایشگاه پایلاتهاده

جلسه ی پنجم
آغازگرها

آغازگرها (Triggers)

- مجموعه ای از کد است که با رخداد رویدادی خاص فعال و اجرا می شود.
- دو نوع SQL Server، Trigger وجود دارد:
 - آغازگرهای DML: با تغییراتی که در داده ها با اعمال درج، حذف و تغییر اعمال می شوند.
 - آغازگرهای DDL: که با تغییرات دیگر، پایگاه داده مانند تعریف و حذف جدول فعال می شوند.

آغازگرها ابزارهای مناسبی برای وآلنس اتوماتیک به برخی از رویدادها هستند.

آغازگرهای DML

- این نوع از آغازگرها برای اعمال حذف، درج و تغییر داده های جداول و view ها قابل تعریف هستند. در تعریف آغازگر می توان تعریف کرد که برای کدام عمل باید فعال شود. بدنه هی آغازگر مانند روال ها از تعدادی دستور T-SQL تعریف می شود.
- این نوع آغازگرها بر دو نوع هستند:
 - نوع AFTER که آغازگر بعد از اعمال تغییر اجرا می شود.
 - نوع INSTEAD OF که آغازگر به جای عملیات تغییر اصلی اجرا می شود.

DML آغازگر

```
CREATE TRIGGER trigger_name  
ON table  
[with <dml_trigger_option>, ...]  
AFTER [INSERT] [,] [UPDATE] [,] [DELETE]  
AS <sql_statement> ...
```

```
ALTER TRIGGER trigger_name ...
```

```
DROP TRIGGER trigger_name
```

```
ENABLE TRIGGER trigger_name ON object_name  
DISABLE TRIGGER trigger_name ON object_name
```

- تعریف این آغازگر به صورت زیر است:

- برای تغییر یک آغازگر از دستور مقابل استفاده می‌گردد:

- برای حذف یک آغازگر:

- آنکه بفواهیم یک آغازگر خاص فعال و یا غیرفعال شود از دستورات زیر استفاده می‌گردد:

مثال

```
create trigger trigger1 on book
after insert as
print 'ok'

insert into book(bookname,BSD) values('HHH',33);

ok
(1 row(s) affected)
```

```
create trigger trigger3 on book
after update as
insert into bookaudit(bookname,BSD) values('newbook',43);
```

آنچه از آنچه در جدول مجازی با نام های **DELETED** و **INSERTED** در اختیار آنهاست قرار می دهد.

اطلاعات جدول <i>deleted</i>	اطلاعات جدول <i>inserted</i>	نوع تغییر
ردیفهای حذف شده	ردیفهای اضافه شده	INSERT
ردیفهای قدیمی (پیش از تغییر)	ردیفهای جدید (پس از تغییر)	UPDATE
ردیفهای حذف شده		DELETE

جدول ۱: محتوای جدول های مجازی

مثال

```
create trigger trigger4 on book  
after insert as  
insert into bookaudit(bookname,BSD)  
select inserted.bookname,inserted.BSD  
from inserted;
```

```
insert into book(bookname,BSD) values ('|||||',12);
```

	id	bookname	BSD
	1	kkk	43
	2	kkk	43
	3	newbook	43
	4	kkk	43
	5		12

- آن بفواهیم تعداد سطرهای تهدت تئثیر قرار گرفته نمایش داده نشود:

```
set nocount on --dont display changed line
```

- در آغازگرها با استفاده از تابع **UPDATE** (که متفاوت از **ستور UPDATE** است) می‌توان کنترل کرد که آیا مقادیر یک ستون خاص جدول تغییر یافته است یا نه (در صورتی که اعمال **insert** و یا **update** بر روی مقادیر این ستون اعمال شود)

غیر فعال کردن فراخوانی تو در توی آغازگرها

- اگر آغازگر **t1** تغییراتی را در جدولی اعمال کند، این تغییرات ممکن است به فعال شدن آغازگر دیگری مانند **t2** منجر شود و این زنجیره می‌تواند همین طور ادامه یابد. برای جلوگیری از این زنجیره‌ها با طول نامحدود در **SQL SERVER** حداقل طول این زنجیره برابر با **32** در نظر گرفته شده است.
- با دستور زیر می‌توان فعال شدن تو در توی **trigger**‌ها را غیر فعال کرد.

```
EXEC sp_configure 'nested triggers', 0  
RECONFIGURE WITH OVERRIDE
```

- اگر به جای عدد صفر، یک قرار دهیم، فراخوانی تو در تو دوباره فعال می‌شود.

-
- برای مشاهده ی **trigger** کرده و درگزینه ی **expand** مورد نظر، از **table trigger** لیستی از **Trigger** ها قابل مشاهده است.

DDL Triggers

- آغازگرهای DDL با عبارات **ALTER**, **CREATE**, **DROP** استفاده می شود.
- این trigger ها بر روی یک دیتابیس تعریف می شوند.
- این trigger ها قابل مشاهده هستند.

دستور کار جلسه ی پنجم

جدول زیر را ایجاد کنید:

Book:

(ID(int, identity, primary key), Bookname(varchar(30)), yearpublish(int), authorname(varchar(40)), QTY(int))

Trigger های زیر را ایجاد کنید. در هر مرحله عکسی از trigger و صحت آن درج کنید.

۱ - trigger بنویسید که در زمان insert کردن داده‌ها، به جای عملیات insert، پیام 'No change was done' را نشان دهد.

۲ - trigger بنویسید که پس از عملیات insert و delete، تمامی مقادیر insert شده و delete شده را در جدول دیگری با نام Book_Audit وارد کند (در جدول Book_Audit فیلد Ins_or_del اضافه کنید که در هنگام del مقدار صفر و در زمان

insert مقدار ۱ بگیرد). [برای عکس گرفتن از نتیجه هم با insert و هم با Delete امتحان کنید]

توجه: در زمان اجرای این trigger1 غیرفعال کنید.

۳ - trigger بنویسید که مانع از update فیلد Bookname شود. [راهنمایی: از rollback و تابع استفاده کنید]

۴ - view را تعریف کنید که نام هر کتاب را به همراه نام نویسنده‌ی آن کتاب و تعدادچاپ(QTY) را نمایش دهد.

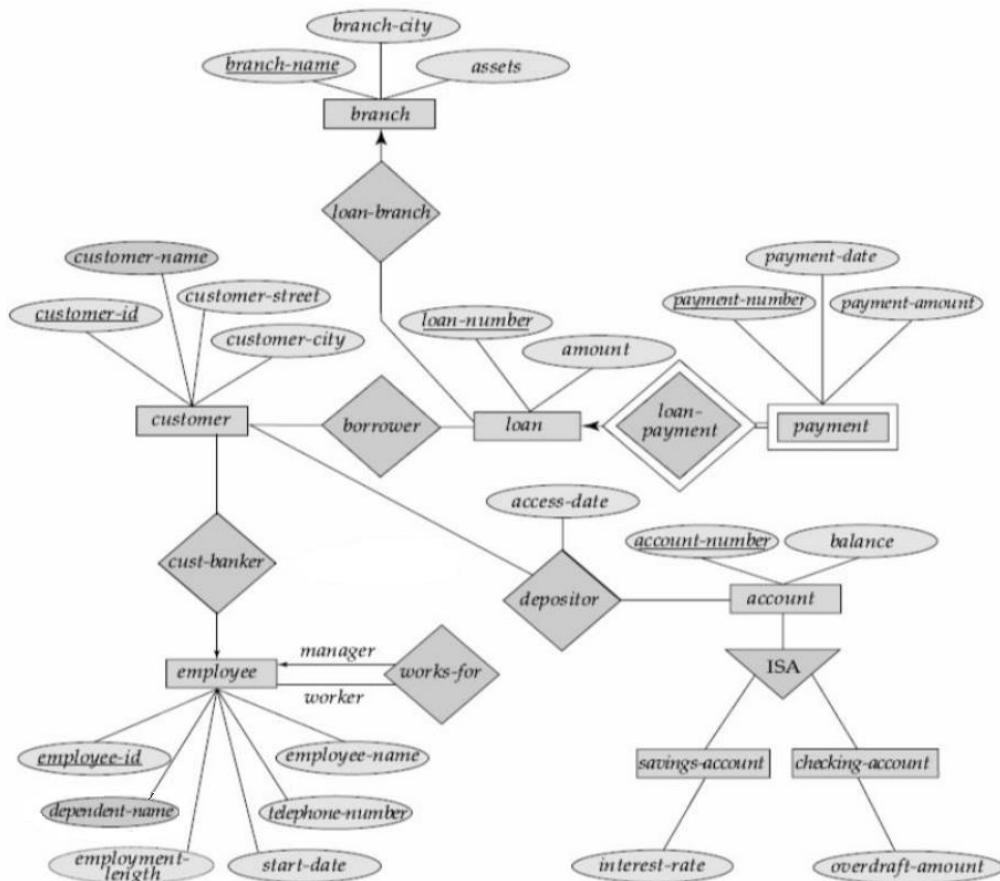
بر روی این view، trigger تعریف کنید که در هنگام insert، کتاب‌هایی را که QTY<1000 دارند را به QTY=1000 تغییر دهد.

۵ - trigger بنویسید که امکان حذف جدول Book را از ما بگیرد.

دستور کار جلسه‌ی ششم

دوره‌ی کلی SQL Server

۱- جداول نمودار ERD زیر را ایجاد کنید، تبدیل هر entity به جدول را توضیح دهید.



۲- پرس و جوهای زیر را با استفاده از T_SQL نوشه و عکس پرس و جو را به همراه نتیجه در گزارشکار بیاورید.

الف) view بنویسید که نام و id مشتری را به همراه میزان وام دریافتی و شعبه‌ی دریافتی او نشان دهد.

ب) شماره حساب و میزان سود این حسابها در صورتی که زمان واریز بعد از سال ۲۰۰۹ باشد.

ج) شماره‌ی بازپرداخت وام‌هایی که شعبه‌ی آن‌ها تهران است.

۲- رویه‌های زیر را ایجاد کنید:

الف) روالی بنویسید که نام مراجعه‌کننده به بانک را به عنوان ورودی بگیرد و *balance* و شماره حساب را به عنوان خروجی برگرداند.

ب) روالی بنویسید که Payment_number را به عنوان ورودی بگیرد و نام شعبه‌ای را که از آن پرداخت صورت گرفته برگرداند.
ج) روالی بنویسید که شماره‌ی مراجعه‌کننده را به عنوان ورودی گرفته و پس از ۱۰ ثانیه اطلاعات مراجعه‌کننده را نمایش دهد.

-۳- توابع زیر را ایجاد کنید:

الف) تابعی بنویسید که شماره حساب، حسابی را که بیشترین سود را داشته برگرداند.
ب) تابعی بنویسید که employee_id را به عنوان ورودی بگیرد و نام دپارتمانی را که کارمند در آن کار می‌کند به عنوان خروجی برگرداند.

-۴- آغازگرهای زیر را بنویسید:

الف) آغازگری بنویسید که از جدول payment, log ایجاد کند.
ب) آغازگری بنویسید که مانع از update شدن فیلد branch_name در جدول Branch شود.

آزمایشگاه پایگاه داده

جلسه‌ی هفتم

نحوه‌ی ایجاد پایگاه داده- جداول- view و تعریف نقش و کاربر در Oracle

ساختن کاربر Admin

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The top navigation bar includes links for Setup, Preferences, Help, Logout, and Database. The Database tab is selected. The main content area is titled "Database Instance: university". The "Server" tab is selected, and the page displays various management sections:

- Storage**: Control Files, Tablespaces, Temporary Tablespace Groups, Datafiles, Rollback Segments, Redo Log Groups, Archive Logs, Migrate to ASM, Make Tablespace Locally Managed.
- Database Configuration**: Memory Advisors, Automatic Undo Management, Initialization Parameters, View Database Feature Usage.
- Oracle Scheduler**: Jobs, Chains, Schedules, Programs, Job Classes, Windows, Window Groups, Global Attributes, Automated Maintenance Tasks.
- Statistics Management**: Automatic Workload Repository, AWR Baselines.
- Resource Manager**: Getting Started, Consumer Groups, Consumer Group Mappings, Plans, Settings, Statistics, Parallel Statement Queue.
- Security**: Users (highlighted with a red box), Roles, Profiles, Audit Settings, Transparent Data Encryption, Oracle Label Security, Virtual Private Database, Application Contexts, Enterprise User Security, Database Vault.
- Query Optimizer**: Manage Optimizer Statistics, SQL Plan Control, SQL Tuning Sets.
- Change Database**: Add Instance, Delete Instance.

مراحل ساختن کاربر admin

Users

Object Type: User

Search
Enter an object name to filter the data that is displayed in your results set.

Object Name: Go

By default, the search returns all uppercase matches beginning with the string you entered. To run an exact or case-sensitive match, double quote the search string. You can use the wildcard symbol (%) in a double quoted string.

Selection Mode: Single

Create

Select	UserName	Account Status	Expiration Date	Default Tablespace	Temporary Tablespace	Profile	Created	User Type
<input checked="" type="radio"/>	ANONYMOUS	EXPIRED & LOCKED	08-Dec-2010 01:52:01 IRST	SYSAUX	TEMP	DEFAULT	08-Dec-2010 01:19:50 IRST	LOCAL
<input type="radio"/>	APEX_030200	EXPIRED & LOCKED	08-Dec-2010 01:52:01 IRST	SYSAUX	TEMP	DEFAULT	08-Dec-2010 01:38:43 IRST	LOCAL
<input type="radio"/>	APPX_PUBLIC_TSFR	EXPIRED & LOCKED	08-Dec-2010 01:52:01 IRST	TSFRS	TEMP	DEFAULT	08-Dec-2010 01:38:43 IRST	LOCAL

Previous 1-25 of 32 Next Create

Database Instance: university > Users >

Create User

General Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

* Name admin

ORACLE Enterprise Manager 11g
Database Control

Setup Preferences Help Logout

Database

Database Instance: university > Users >

Logged in As SYS

Create User

Show SQL Cancel OK

General Roles System Privileges Object Privileges Quotas Consumer Group Privileges Proxy Users

Edit List

Role	Admin Option	Default
CONNECT	<input type="checkbox"/>	<input checked="" type="checkbox"/>

ORACLE Enterprise Manager 11g
Database Control

Setup Preferences Help Logout

Database

Database Instance: university > Users >

Logged in As SYS

Modify Roles

Cancel OK

Available Roles

JAVADEBUGPRIV
JAVAIDPRIV
JAVASYSPRIV
JAVAUSERPRIV
JAVA_ADMIN
JAVA_DEPLOY
JMXSERVER
LOGSTDBY_ADMINISTRATOR
MGMT_USER
OEM_Advisor

Selected Roles

CONNECT
DBA
LBAC_DBA

>
Move
>>
Move All
<
Remove
<<
Remove All

ایجاد جداول



پس از آن با یوزر admin وارد می شویم:

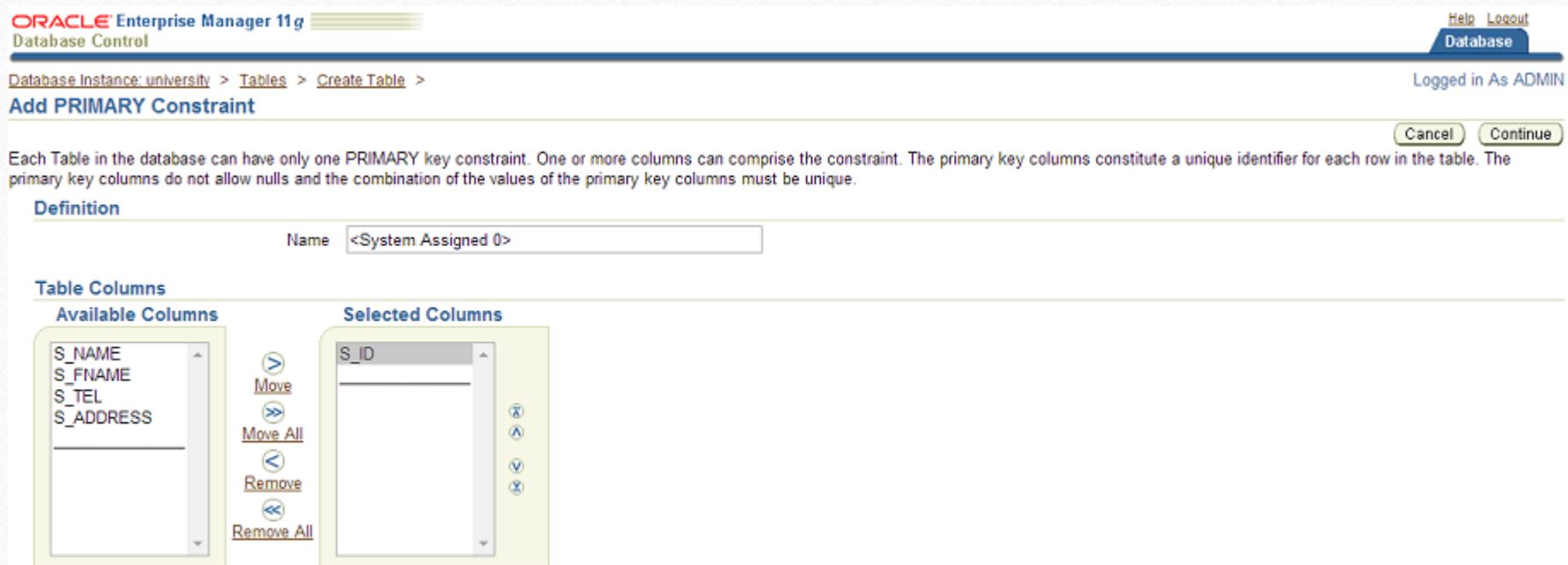
The screenshot shows the Oracle SQL Developer interface for creating a new table named 'ADMIN.STUDENT'. The 'General' tab is selected, showing details like Name: STUDENT, Schema: ADMIN, Tablespace: SYSTEM, and Organization: Standard (Heap Organized). The 'Columns' tab displays the structure of the table with columns S_ID, S_NAME, S_FNAME, S_TEL, S_ADDRESS, and S_DEP, each with their respective data types and constraints.

Column	Name	Data Type	Size	Scale	Not NULL	Default Value	Encrypted
1	S_ID	NUMBER			<input checked="" type="checkbox"/>		<input type="checkbox"/>
2	S_NAME	VARCHAR2	20		<input type="checkbox"/>		<input type="checkbox"/>
3	S_FNAME	VARCHAR2	30		<input type="checkbox"/>		<input type="checkbox"/>
4	S_TEL	NUMBER			<input type="checkbox"/>		<input type="checkbox"/>
5	S_ADDRESS	VARCHAR2	70		<input type="checkbox"/>		<input type="checkbox"/>
6	S_DEP	VARCHAR2	30		<input type="checkbox"/>		<input type="checkbox"/>

پس از آن در تب General فیلدهای جدول را وارد میکنیم:

ایجاد constraint ها

- برای تعیین کلید اصلی و سایر محدودیت ها در تب constraints، ابتدا نوع محدودیت را انتخاب کرده و add را می زنیم:



ادامه ی ساختن جداول

- در نهایت با فشردن دکمه ی **Create** جدول ساخته می شود:

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The title bar reads "ORACLE Enterprise Manager 11g Database Control". The top menu includes "Help", "Logout", and "Database". A status bar at the top right says "Logged in As ADMIN". The main area is titled "Tables" and shows a confirmation message: "Table ADMIN.STUDENT has been created successfully". Below this, there is a search section with fields for "Schema" (set to "ADMIN") and "Object Name", and a "Go" button. A note below the search fields explains the search behavior. Under "Selection Mode", "Single" is selected. At the top of the table list, there are buttons for "Edit", "View", "Delete With Options", "Actions", "Create Like", and "Go". The table list itself has columns: "Select", "Schema", "Table Name", "Tablespace", "Partitioned", and "Rows Last Analyzed". One row is visible, showing "ADMIN" in the Schema column and "STUDENT" in the Table Name column. A "Create" button is located at the top right of the table list.

تعريف نقش ها

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. At the top, it displays "ORACLE Enterprise Manager 11g" and "Database Control". On the right, it shows "Logged in As ADMIN". The main menu bar includes "Home", "Performance", "Availability", "Server" (which is selected), "Schema", "Data Movement", and "Software and Support". Below the menu, there are several sections: "Storage" (Control Files, Tablespaces, Temporary Tablespace Groups, Datafiles, Rollback Segments, Redo Log Groups, Archive Logs, Migrate to ASM, Make Tablespace Locally Managed); "Database Configuration" (Memory Advisors, Automatic Undo Management, Initialization Parameters, View Database Feature Usage); "Oracle Scheduler" (Jobs, Chains, Schedules, Programs, Job Classes, Windows, Window Groups, Global Attributes, Automated Maintenance Tasks); "Statistics Management" (Automatic Workload Repository, AWR Baselines); "Resource Manager" (Getting Started, Consumer Groups, Consumer Group Mappings, Plans, Settings, Statistics, Parallel Statement Queue); and "Security" (Users, Roles, Profiles, Audit Settings, Transparent Data Encryption, Oracle Label Security, Virtual Private Database, Application Contexts, Enterprise User Security, Database Vault). The "Roles" link under the Security section is highlighted with a red box.

← → ⌂ https://localhost:1158/em/console/database/security/role?target=university&type=oracle_database&cancelURL=/em/console/database/databaseObjectsSearch ⋮

ORACLE Enterprise Manager 11g Database Control Database Help Logout Database Logged in As ADMIN

Database Instance university > Roles > Create Role

Show SQL Cancel OK

General Roles System Privileges Object Privileges Consumer Group Privileges

* Name

Authentication

* Enter Password

* Confirm Password

For Password choice, the role is authorized via password.

General Roles System Privileges Object Privileges Consumer Group Privileges

تعريف view

The screenshot shows the Oracle Enterprise Manager 11g Database Control interface. The top navigation bar includes links for Help, Logout, and Database. It also indicates that the user is Logged in As ADMIN. The main menu has tabs for Home, Performance, Availability, Server, Schema (which is selected), Data Movement, and Software and Support.

In the Schema section, under Database Objects, the Views link is highlighted with a red box. Other options include Tables, Indexes, and Synonyms.

Under Programs, the Options include Packages, Package Bodies, Procedures, and Functions.

Under Materialized Views, the Options include Materialized Views, Materialized View Logs, Refresh Groups, and Dimensions.

The central workspace shows the creation of a view named PERSONAL_VIEW. The General tab is selected. The view details are as follows:

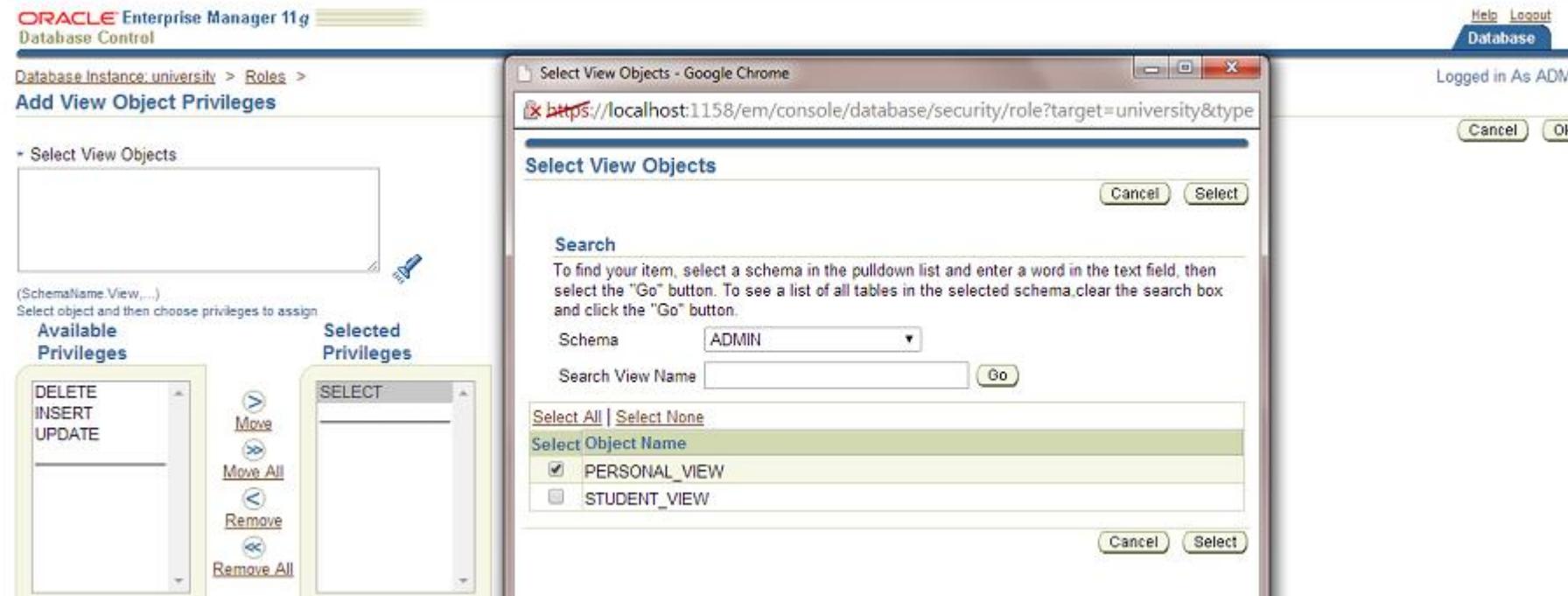
- Name: PERSONAL_VIEW
- Schema: ADMIN
- Aliases: "S_ID", "S_NAME", "S_FNAME", "S_TEL", "S_ADDRESS", "S_DEP"
- Status: VALID
- * Query Text:

```
SELECT "S_ID", "S_NAME", "S_FNAME", "S_TEL", "S_ADDRESS", "S_DEP"
FROM STUDENT
WHERE STUDENT.S_ID=(SELECT SYS_CONTEXT(USERENV,'SESSION_USER')
SESSION_USER FROM DUAL)
```

Action buttons at the bottom right include Create Like, Go, Show SQL, Revert, and Apply.

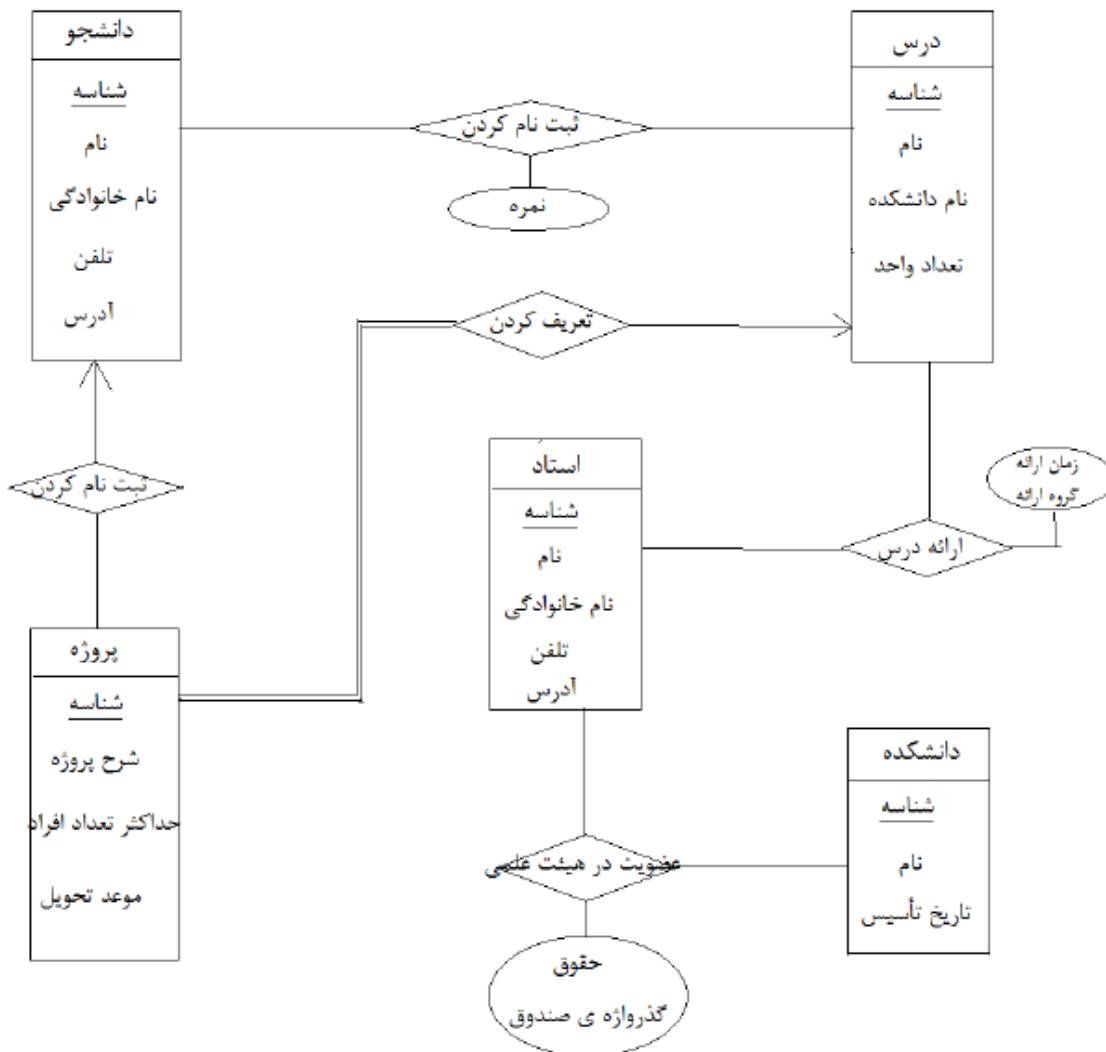
دادن حق دسترسی به نقش ها

دادن حق دسترسی view ایجاد نظر



جلسه‌ی هفتم

- ۱- کاربر admin را ایجاد کنید(مراحل ایجاد آن را توضیح داده و از مراحل عکس بگیرید).
- ۲- با کاربر admin در پایگاه داده وارد شده و جداول زیر را وارد کنید: (مراحل ایجاد آن را توضیح داده و از مراحل عکس بگیرید).



student	(S_ID,S_NAME,S_FNAME,S_TEL,S_ADDRESS,S_DEP)
REGISTER	(GRADE,S_ID,C_ID)
COURSE	(C_ID,C_NAME,C_DEP,C_NUM)
project	(C_ID,PROJ_ID,PROJ_DEF,PROJ_MEMBER,PROJ_DEADLINE)
PROJECT_REG	(S_ID,C_ID,PROJ_ID)
PRESENT	(T_ID,C_ID,CLASSTIME,CLASSGROUP)
MEMBER	(DEP_ID,T_ID,SALARY,PASSWORD)
TEACHER	(T_ID,T_NAME,T_FNAME,T_TEL,T_ADDRESS)
DEPARTMENT	(DEP_ID,DEP_NAME,DEP_DATE)
PUBLICATION	(T_ID,BOOK_NAME,PUBLISH_DATE)

۳- نقش‌های زیر را با دسترسی‌های عنوان شده ایجاد کنید: (مراحل ایجاد آن را توضیح داده و از مراحل عکس بگیرید).

(الف) نقش دانشجو

- ۱- خواندن کلیه‌ی اطلاعات شخصی خود و به روز رسانی آدرس و شماره تلفن خود
- ۲- مشاهده‌ی نمرات و معدل خود

(ب) نقش استاد

- ۱- به روز رسانی تلفن و آدرس خود
- ۲- مشاهده‌ی دروس ارائه شده در دانشکده‌ای که وی عضو هیئت علمی آن است

(ج) نقش مدیر آموزش

دسترسی خواندن، به روز رسانی، اضافه و حذف جدول دانشجویان

۴- برای هر یک از نقش‌های ایجاد شده یک کاربر تعریف کنید(مراحل ایجاد آن را توضیح داده و از مراحل عکس بگیرید).

آزمایشگاه پایگاه داده

جلسه‌ی هشتم

رمزنگاری ستون - روال‌های ذخیره‌شده در Oracle

رمزنگاری ستون



برای رمزنگاری ستون، نیاز به یک کلید داریم. ▶

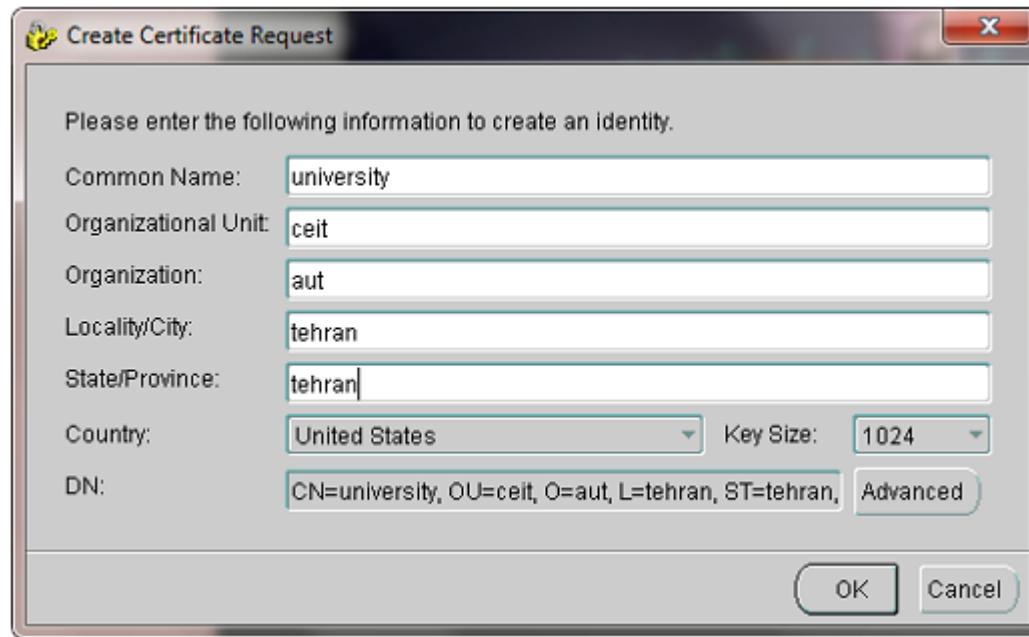
برای ساختن wallet، از wallet manager استفاده می شود: ▶

:wallet ایجاد ▶

1- انتخاب پسورد برای :wallet

مراحل ساختن :wallet

2- ایجاد certificate جدید:



- 3- با کاربر SYS به قسمت server-> transparent data encryption می رویم.
- 4- در مسیر wallet location یک فolderی با نام wallet می سازیم.
- 5- در sql plus با دستور system@orcl وارد می شویم.

مراحل ساختن :wallet

6- با دستور زیر wallet را فعال می کنیم:

```
SQL> alter system set encryption key identified by "sogand2321370";
System altered.
```

7- در تب open به صورت server->transparent data encryption wallet تغییر حالت داده است.

▶ پس از آن با کاربر admin وارد شده و ستون دلخواه را رمز می کنیم.

procedure‌ها

✓ امکان ثبت نام در دروس دانشکده‌ی خود:

Name REGISTER PROCEDURE

Schema ADMIN

Status Valid

Source

```
(  
    C_ID_IN in INTEGER  
)  
as  
mid varchar(20);  
sdep varchar(30);  
cdep varchar(30);  
cursor S1 is  
SELECT SYS_CONTEXT(USERENV,'SESSION_USER') SESSION_USER FROM DUAL;  
cursor S2 is  
SELECT S_DEP FROM STUDENT WHERE S_ID=(SELECT SYS_CONTEXT(USERENV,'SESSION_USER') SESSION_USER  
cursor S3 is  
SELECT C_DEP FROM COURSE WHERE C_ID_IN=C_ID;  
  
begin  
OPEN S1;  
fetch S1 into mid;  
OPEN S2;  
fetch S2 into sdep;  
OPEN S3;  
fetch S3 into cdep;  
  
if sdep=cdep then  
    INSERT INTO REGISTER(GRADE,S_ID,C_ID)VALUES(null,TO_NUMBER(mid,'99999999'),C_ID_IN);  
end if;  
end;
```

```
SQL> execute admin.register_procedure(7777);  
PL/SQL procedure successfully completed.
```

دستور کار جلسه‌ی هشتم

- 1- جداولی را که در جلسه‌ی قبل ایجاد کردید، مقدار دهی کنید.
- 2- پسورد را از قسمت ایجاد ROLE‌ها حذف کنید، سپس اختیارات نقش‌هایی را که در جلسه‌ی گذشته ایجاد کردید، ارزیابی کنید.
- 3- فیلد password از جدول Member را رمز کنید.
- 4- در نقش‌های جلسه‌ی قبل تغییرات زیر را اعمال کنید:
 - (الف) نقش دانشجو: امکان اخذ پروژه‌ی جدید از دروس اخذ شده به شرطی که قبلاً بیشینه‌ی افرادی که می‌توانند ثبت نام نمایند، پر نشده باشد.
 - (ب) نقش دانشجو: امکان حذف دروس اخذ شده قبل از هفته‌ی دوازدهم(فرض کنید هفته‌ی شروع را به عنوان وردی به procedure می‌دهیم).
 - (ج) نقش مدیر سیستم: کلیه‌ی حقوق دسترسی روی کلیه‌ی جداول به جز دسترسی به رمز صندوق استاد.

تمرین‌ها و پروژه

تمرین سری اول (SQL Server)

تمرین ۳ نمره دارد.

تمرین به صورت فردی است.

هر مرحله باید توضیح و شکل مربوطه را شامل شود.

۱- جداول زیر را با constraint های ذکر شده، ایجاد کرده و دیاگرام جداول را رسم کنید.(اعمال constraint را همراه با عکس باشد)

:Patient جدول

ID(int) identity	Patient_name varchar(30)	Zip code (int, Unique)	Doctor_name (varchar(30))(foreign key)	Problem (بیماری) Varchar(40)
------------------	--------------------------	------------------------	--	------------------------------

:Doctor جدول

Doctor_name (varchar(30) primary key)	DEPT_name (varchar(30),unique)	Specialty (varchar(30))	Salary(int) Constraint1:Salary>5000
---------------------------------------	--------------------------------	-------------------------	-------------------------------------

۲- رویه(store procedure) بنویسید که نام و ID یک بیمار را به عنوان ورودی بگیرد. در صورتی که این بیمار در جدول باشد، نام دکتر و تخصص پزشک را نشان بدهد، در صورتی که نه نام و نه ID بیمار در جدول باشد، این فرد را در جدول وارد کند(Zip code) و نام پزشک را به دلخواه در نظر بگیرید).

۳- الف) تفاوت های stored procedure با function را بنویسید.

ب) تابعی(function) بنویسید که ID بیمار را به عنوان ورودی بگیرد و تخصص پزشک معالج را به عنوان خروجی برگرداند.

۴- trigger تعریف کنید که در زمان insert و delete شدن داده‌ها، داده‌های delete شده را در جدول Log_delete وارد کرده و داده‌های insert شده را در جدول Log_insert وارد کند.

سؤال امتیازی: ستون problem از جدول Patient را با استفاده از رمزنگاری متقارن(Symmetric key) رمز کرده، مراحل را توضیح دهید. پس از آن نحوه‌ی بازیابی اطلاعات این ستون را نشان دهید.(0.75)

تمرين اختياری(Oracle):

تمرين انفرادي بوده و ۲ نمره دارد.

پس از تاریخ تعیین شده تحويل گرفته نمی شود.

برای هر قسمت توضیحات کافی به همراه عکس از هر مرحله در گزارش ارائه دهید.

۱- ابتدا کاربر admin را ایجاد کنید.

۲- جداول زیر را ایجاد کنید(کلیداصلی و کلیدهای خارجی را مشخص کنید)

مراجعین بانک	<u>کد ملی</u> - نام خانوادگی- شماره حساب-شماره کارت- ماندهی حساب
حساب بانکی	نوع حساب(بلند مدت، کوتاه مدت و....)- <u>شماره حساب</u> - شماره کارت- تاریخ باز شدن حساب
کارمند بانک	<u>شماره کارمندی</u> - <u>شماره صندوقی</u> که درش کار می - کند- شماره حساب هایی که باز کرده است

۳- کاربری با username=11123456 ایجاد کنید که نقش مراجعه کننده را داشته باشد:

- نقش مراجعه کننده باید قابلیت های زیر را داشته باشد:

- مشاهده اطلاعات حساب خود + نوع حساب و تاریخ باز کردن حساب خود (این مراجعه کننده نباید قادر به مشاهده اطلاعات حساب سایر مراجعین باشد)

۴- فیلد شماره حساب در جدول حساب بانکی را رمز کنید.

۵- کاربری با username=99923456 ایجاد کنید که نقش کارمند را داشته باشد.

- نقش کارمند باید قابلیت های زیر را داشته باشد:

- امكان مشاهده، اضافه و حذف کردن شماره حساب ها (یک شماره حساب را به عنوان ورودی می گیرد، متغیری از نوع int را هم به عنوان ورودی دریافت می کند، در صورتی که این ورودی صفر باشد شماره حساب مربوطه را حذف می کند و در صورتی که این مقدار یک باشد، شماره حساب را اضافه می کند)

پروژه‌ی پایانی آزمایشگاه دیناتبیس (Oracle)

- پروژه به صورت گروهی(گروههای دو نفره) انجام می‌گیرد.
- تحويل حضوری پروژه به صورت انفرادی خواهد بود.
- تحويل حضوری شامل تست عملی موارد خواسته شده خواهد بود.

پایگاه داده مربوط به شرکت بیمه است؛ طراحی آن باید بر اساس نیازمندی‌های زیر باشد(طراحی پایگاهداده:۷۵،۰،۰ نمره و ایجاد جداول ۷۵،۰ نمره)

- بیمه گزار دارای یک شناسه‌ی یکتا است و اطلاعات مربوط به نام، نام خانوادگی، تلفن و آدرس وی در پایگاه داده ذخیره می‌شود.
هر بیمه گزار می‌تواند در بیمه‌های مختلفی ثبت نام کند(هر نوع قرارداد بیمه تنها یکبار می‌تواند توسط بیمه گزار اخذ شود مگر این که قرارداد یک ساله به پایان رسیده باشد)
- هر بیمه‌گر دارای یک شناسه‌ی کارمندی یکتا است و اطلاعات مربوط به نام، نام خانوادگی، تلفن و آدرس وی در پایگاه داده ذخیره می‌شود.
- هر نوع بیمه با یک شناسه‌ی یکتا مشخص می‌گردد. اطلاعات مربوط به نام بیمه و مبلغ لازم جهت قرارداد در پایگاه داده ذخیره می‌شود.
- بیمه‌ها انواع مختلفی دارند(بیمه‌ی اموال، بیمه‌ی اشخاص، بیمه‌ی مسئولیت و بیمه‌های خاص) برای بیمه‌ی اشخاص و بیمه‌ی اموال، بیمه‌شده نیز در پایگاه داده ذخیره می‌شود.
- هر نوع بیمه ممکن است در شعب مختلفی ارائه شود. شعب مختلف نام و شماره‌ی شعبه دارند.
- هر شعبه، یک رئیس شعبه و یک معاون امور مالی دارد.
- هر کارمند بیمه در هر شعبه‌ای که مشغول به کار است، یک صندوق دارد. استفاده از این صندوق نیازمند وارد کردن یک عدد به عنوان گذر واژه است.
- مبلغ پرداختی توسط بیمه گزار برای هر نوع بیمه قراردادی در پایگاه داده ذخیره می‌شود.

نقش‌هایی که در پایگاه داده وجود دارند، به قرار زیر است:

- ✓ بیمه گزار
- ✓ بیمه گر
- ✓ مدیر شعبه
- ✓ مدیر سیستم

اعمالی که هریک از نقش‌ها می‌توانند بر روی پایگاه داده انجام دهند، به شرح زیر است(در هر بخش در صورتی که فرضیات جدید مورد نیاز است، با ذکر آن‌ها می‌توان تغییراتی را در جداول اعمال کرد):

✓ بیمه گزار:

- خواندن کلیه‌ی اطلاعات شخصی خود و به روزرسانی آرس و شماره تلفن خود.

- مشاهده‌ی انواع بیمه‌ای که ارائه می‌گردد.
 - مشاهده‌ی بیمه‌هایی که در آن‌ها ثبت نام کرده است، به همراه تاریخ شروع قرار داد.
 - امکان اخذ بیمه‌ی جدید به شرطی که قبلاً در آن ثبت نام نکرده باشد و یا قرار داد یک‌ساله به پایان رسیده باشد.
 - امکان لغو قرار داد قبل از هفته‌ی چهارم و با کسر $\frac{1}{5}$ هزینه بابت جریمه‌بابت لغو قرار داد.
- ✓ بیمه‌گر:
- خواندن کلیه‌ی اطلاعات شخصی خود و گذرواژه‌ی صندوق خود.
 - امکان به روزسانی تلفن، آدرس و گذرواژه‌ی دسترسی به صندوق شخصی.
 - امکان عقد قرارداد بیمه با بیمه‌گزار.
- ✓ مدیر شعبه:
- استخدام کارمندان.
 - اضافه و حذف اطلاعات مربوط به جدول بیمه.
 - از آنجایی که مدیر شعبه خود نیز کارمند است، کلیه‌ی حقوق بیمه‌گر را دارد.
- ✓ مدیر سیستم:
- کلیه‌ی حقوق دسترسی بر روی کلیه‌ی جداول به جز دسترسی به گذرواژه‌ی بیمه‌گر
- برای هریک از نقش‌ها کاربری ایجاد کنید و درستی اعمال نقش‌ها را نشان دهید.(۳ نمره اعمال خط مشی‌ها و ایجاد کاربران)

پایگاه داده باید امکانات زیر را نیز داشته باشد:

- گذرواژه‌ی صندوق بیمه‌گر به صورت رمز شده در پایگاه داده ذخیره می‌شود.(۰,۵ نمره)
- کاربری به نام dean در پایگاه داده وجود دارد که مدیر شعبه است و تمامی اعمال او audit می‌شود.(۰,۵ نمره)
- با اتمام مدت یک‌ساله‌ی قرارداد، قرارداد خودبه‌خود لغو می‌گردد.(۰,۵ نمره)

بخش اختیاری(یک نمره)

اینترفیسی برای نقش بیمه‌گر طراحی نمایید که از طریق وب بتوان با پایگاه‌داده ارتباط برقرار کرد.

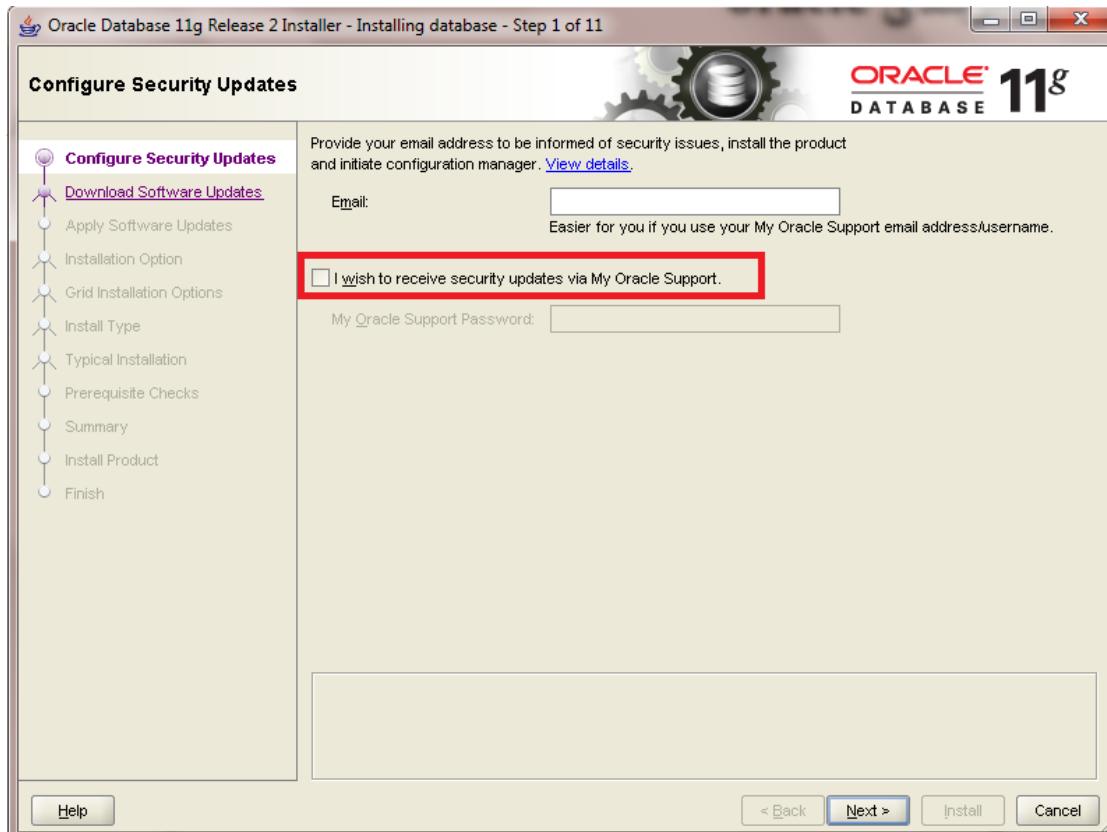
پیوست‌ها

- راهنمای نصب دیتابیس oracle

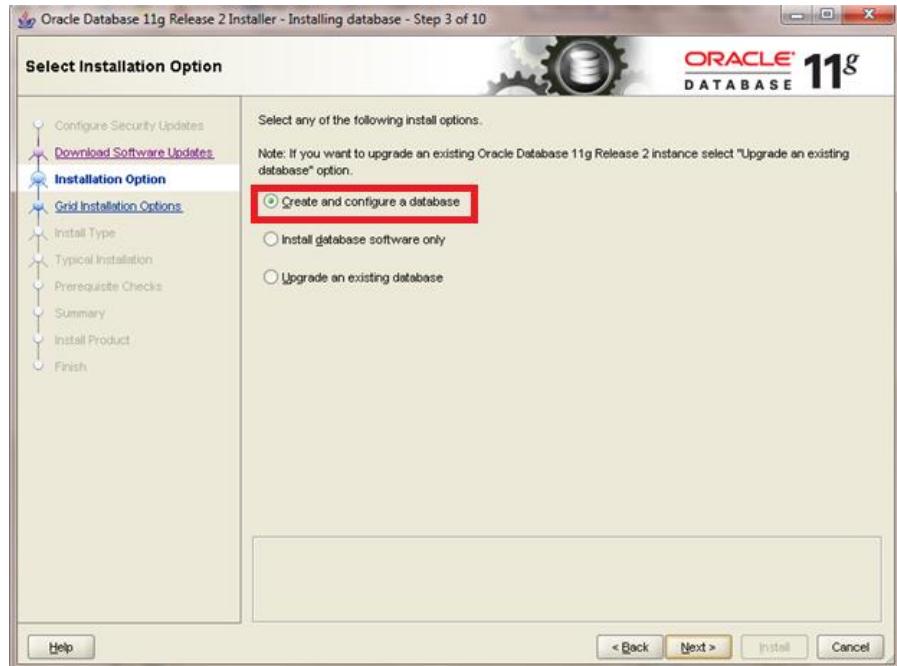
مراحل مهم در ادامه بیان شده است:

پس از فشردن دکمه‌ی setup، مراحب نصب به صورت زیر انجام می‌گیرد.

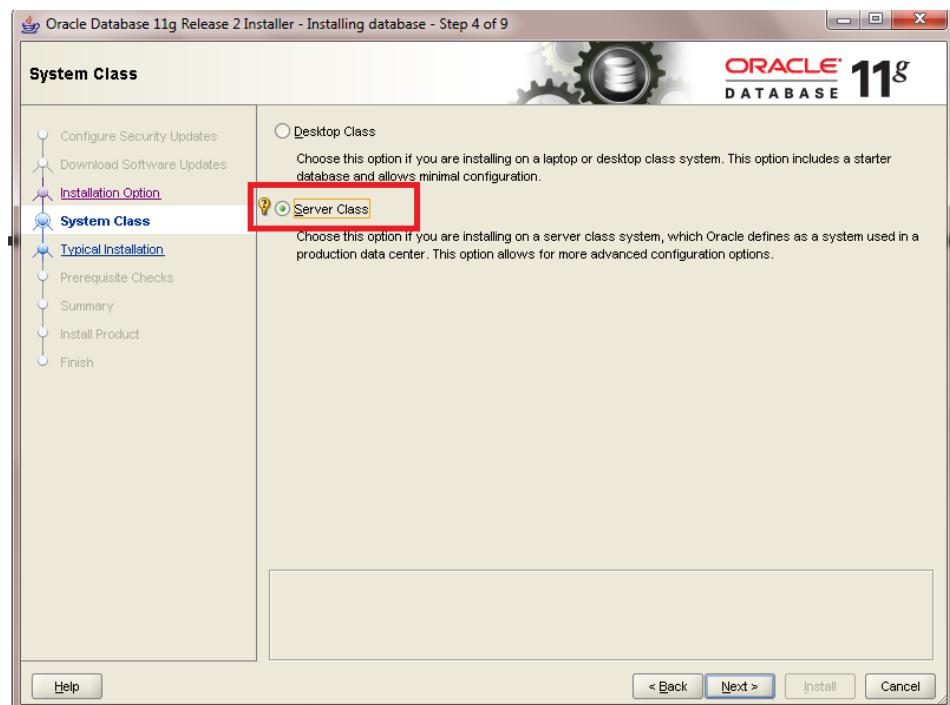
- در اولین مرحله درصورتی که مایل به دریافت بهروزرسانی‌های امنیتی هستیم ایمیل و پسورد ورودی به سایت oracle را وارد می‌کنیم.



-۲

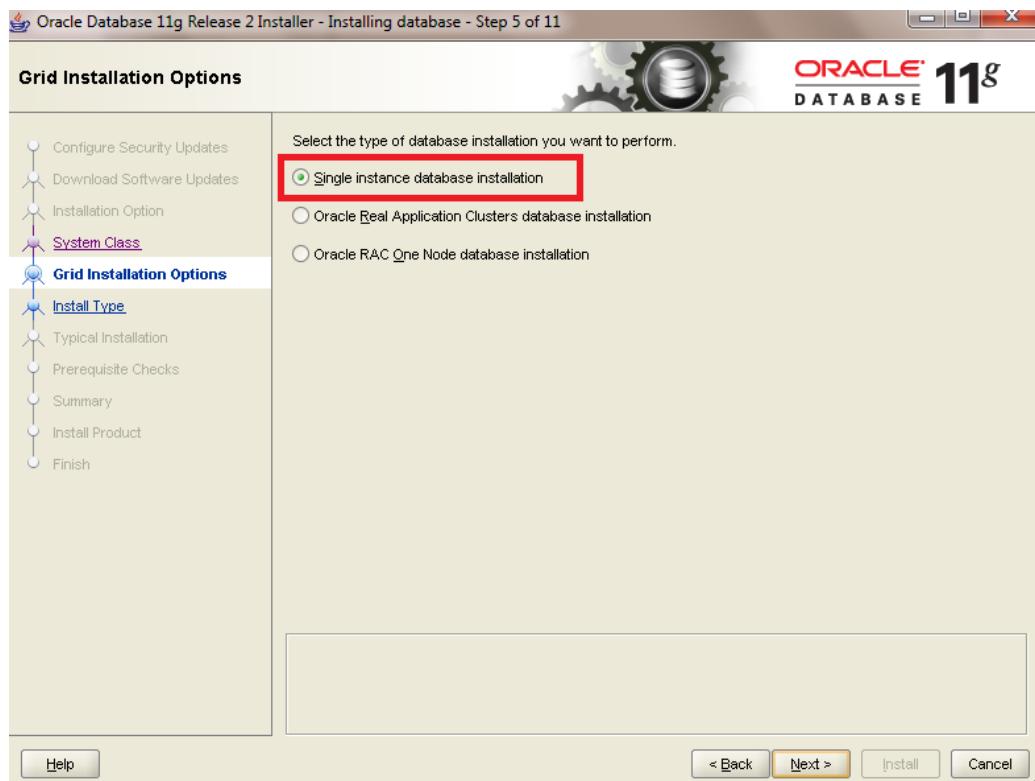


-۳

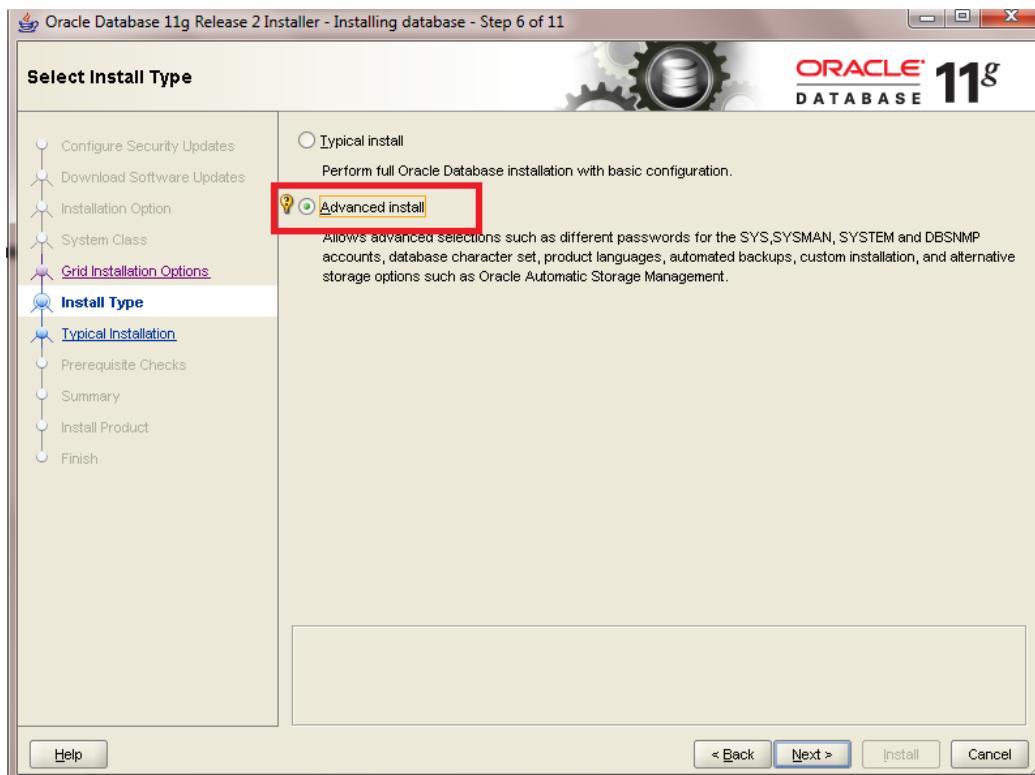


۳

-۴

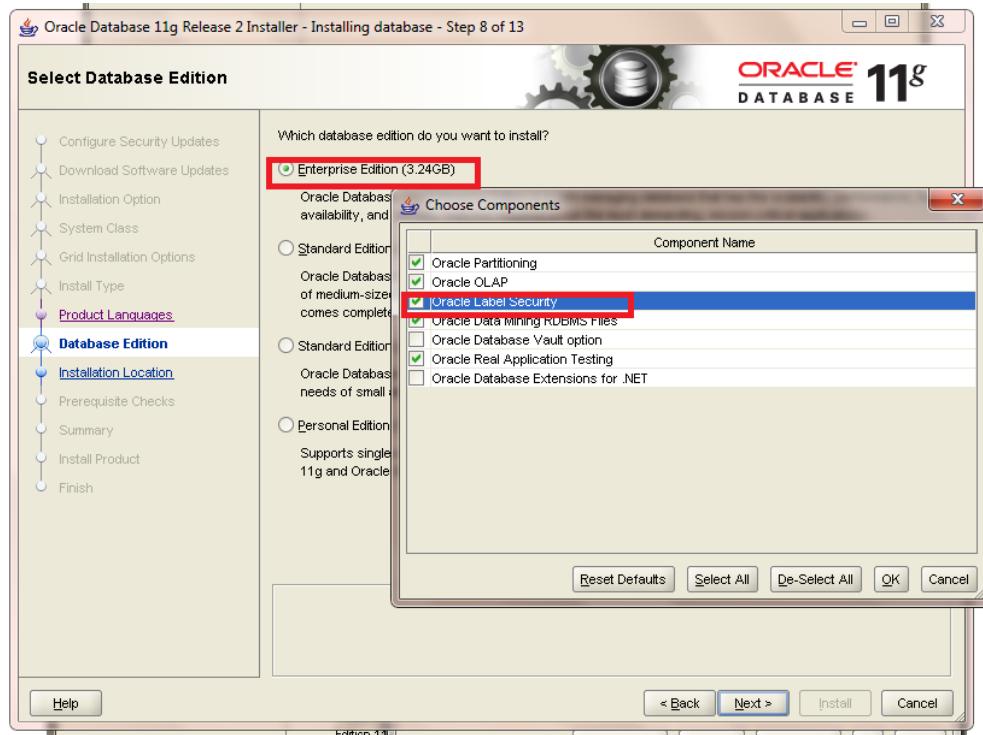


-۵

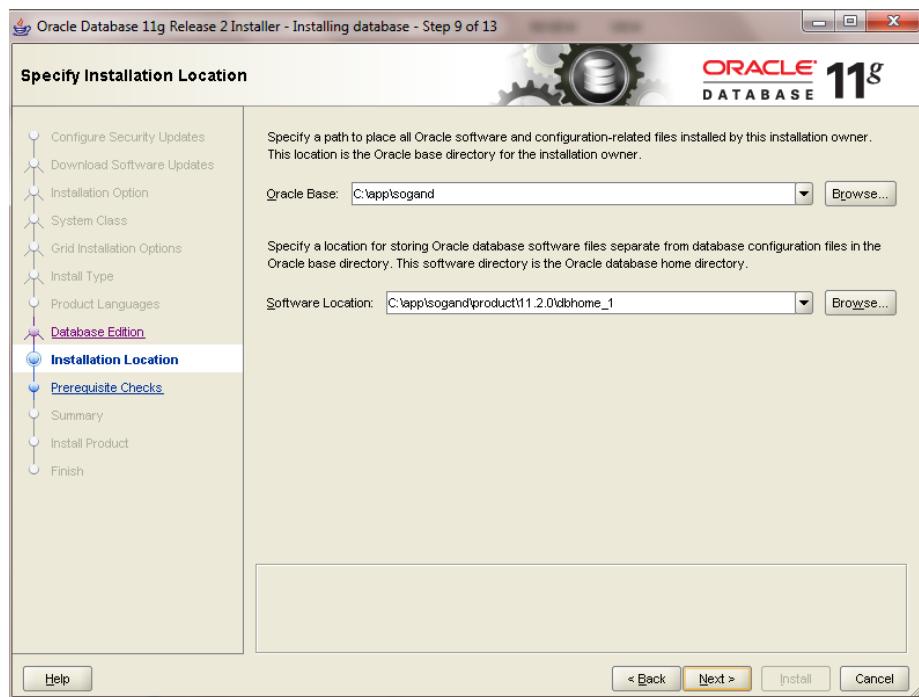


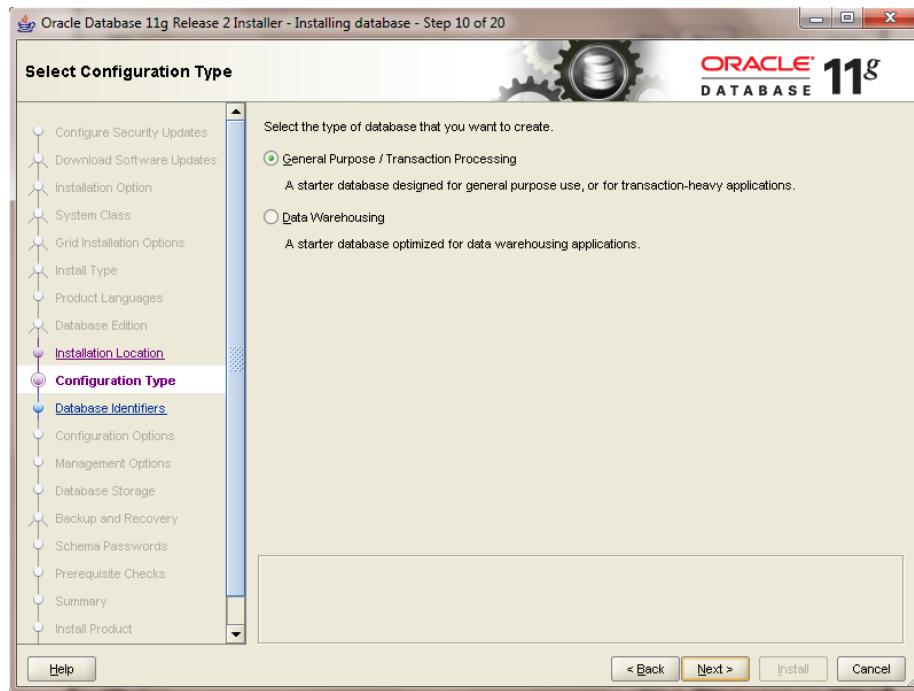
۳

۶- در این مرحله مشخص می‌کنیم که امکان استفاده از برچسب‌های امنیتی ایجاد گردد.

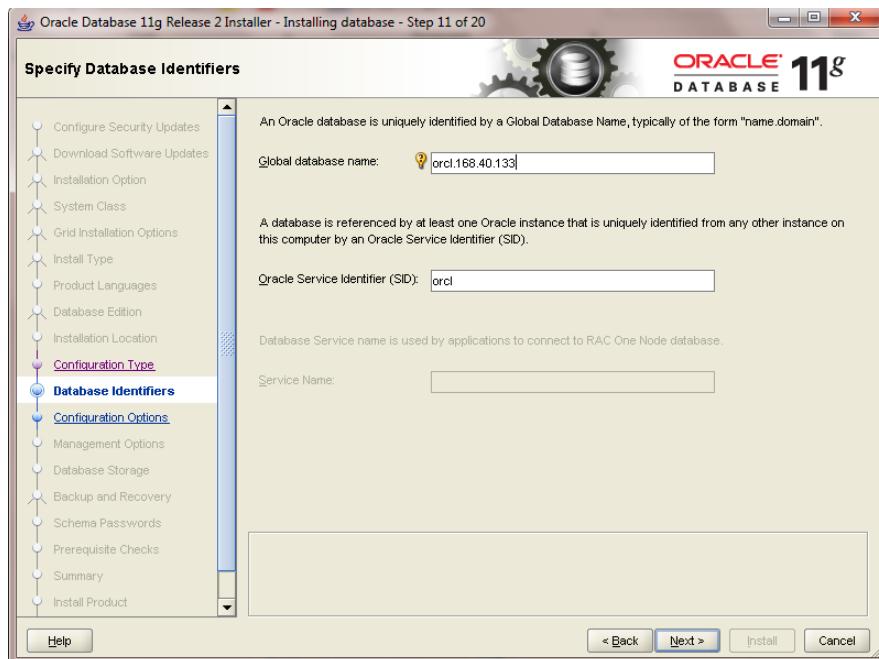


۷- در این مرحله، مکانی که برنامه قرار است نصب شود، مشخص می‌گردد. نکته‌ی مهم برای تعیین این محل این است که
نباید در مشخص کردن این مکان فاصله وجود داشته باشد.

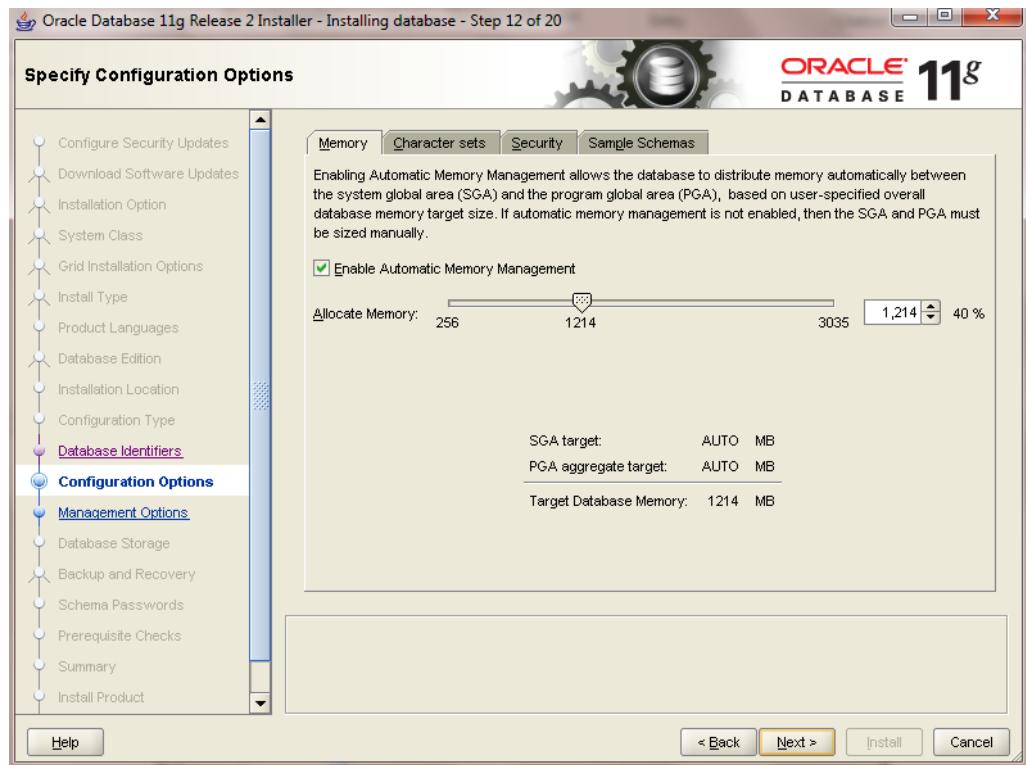




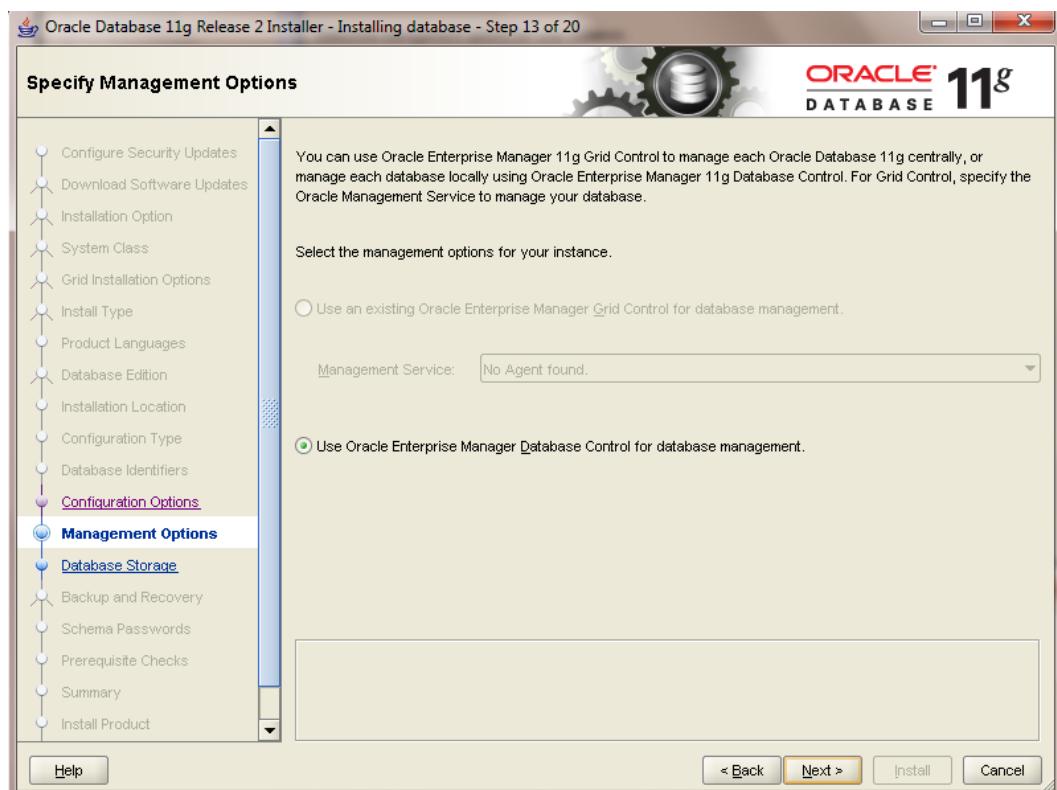
۹- در این مرحله نام دیتابیس مشخص می‌شود. بنابراین یک نام دلخواه به عنوان نام دیتابیس وارد می‌گردد:

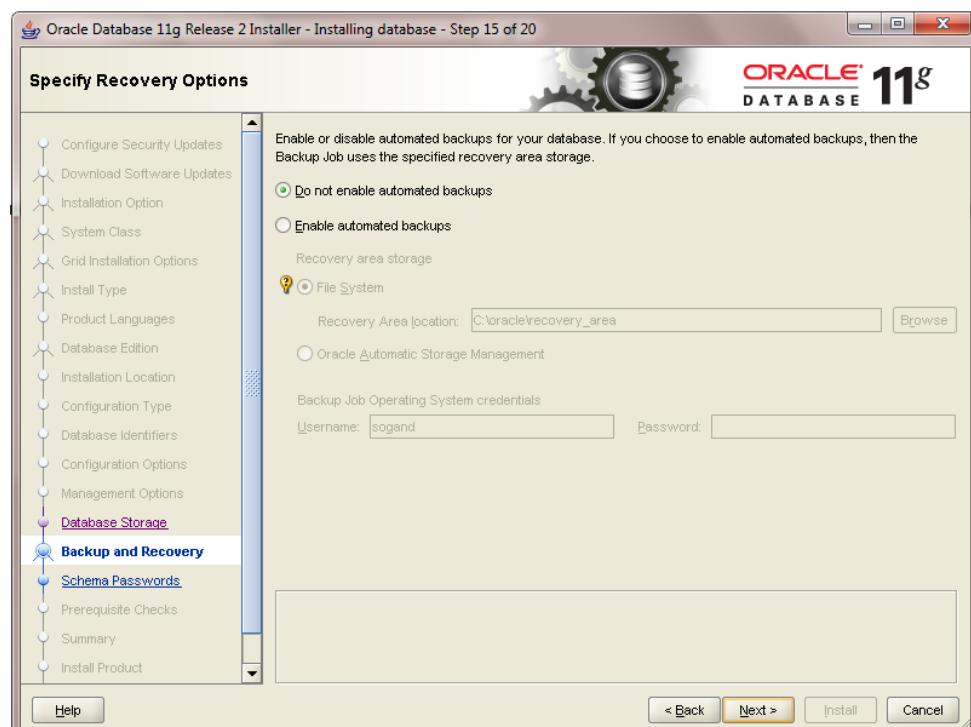
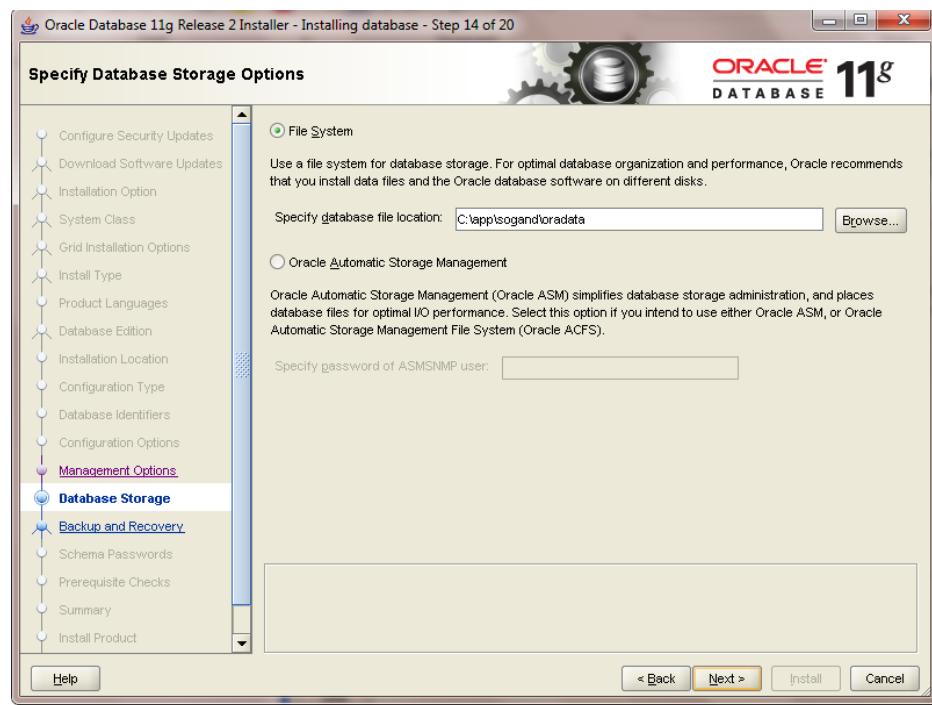


۱۰- در این مرحله بخشی از حافظه (میزان فضای لازم) که قرار است در اختیار دیتابیس قرار بگیرد، مشخص می‌گردد.

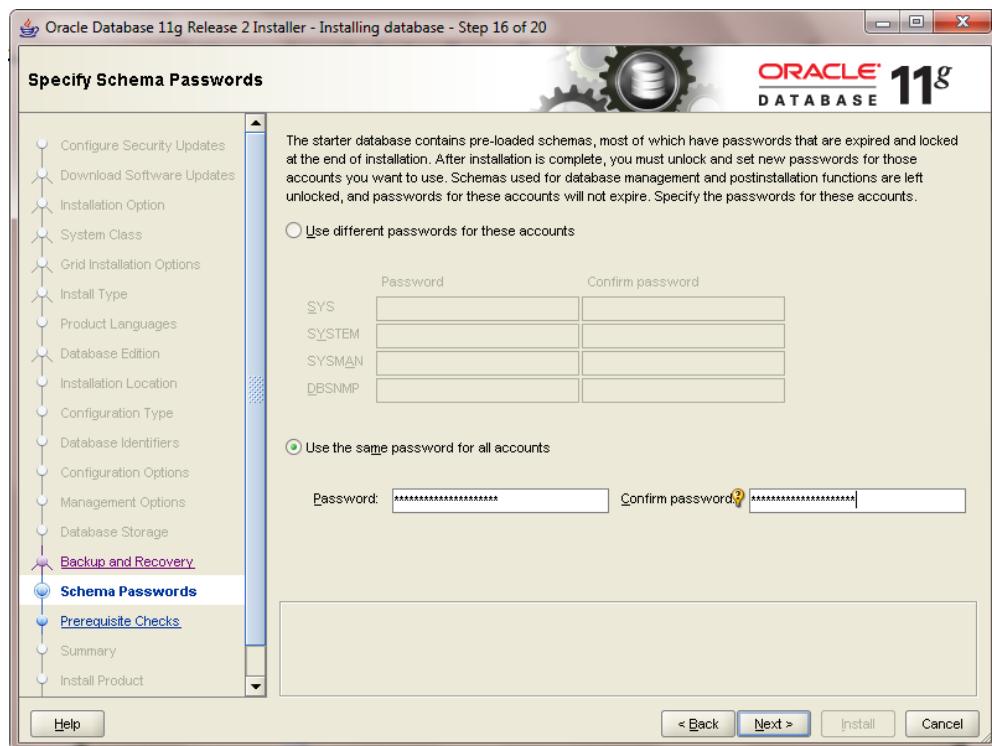


- ۱۱

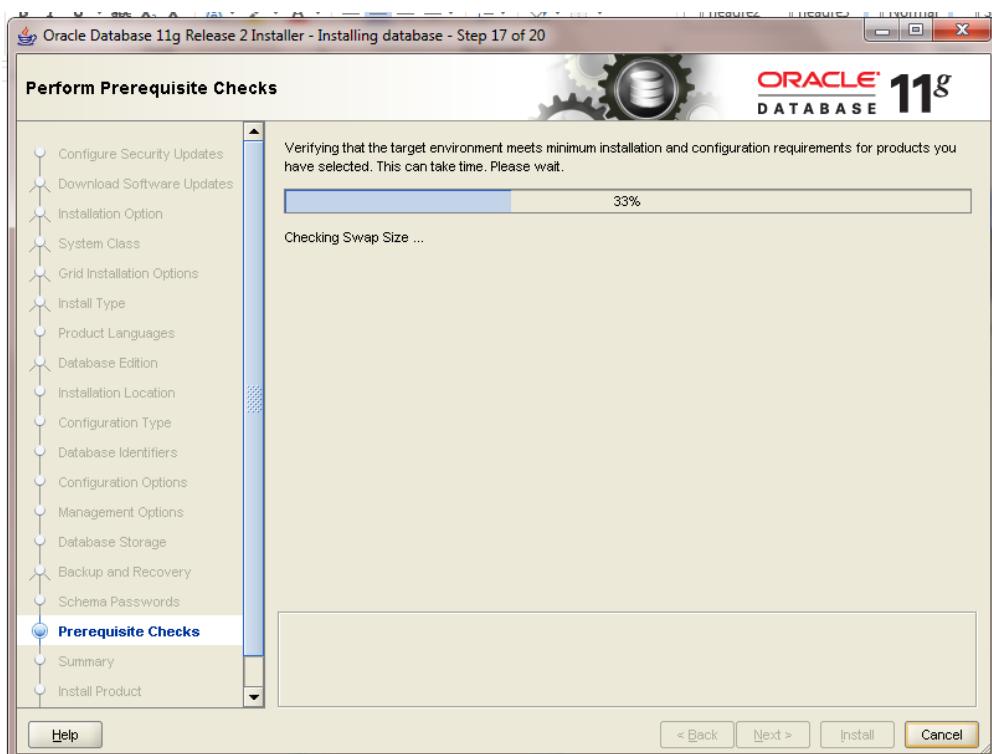




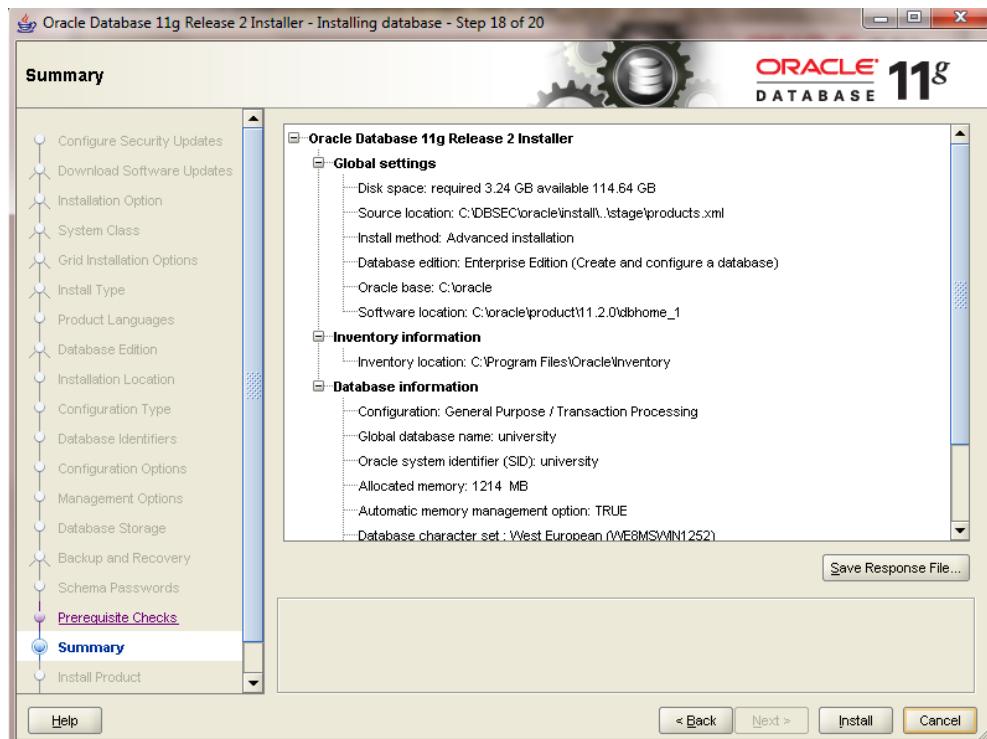
۱۴- در این مرحله باید یوزرنیم و پسورد برای دیتابیس مورد نظر انتخاب گردد. به این منظور باید تنها از حروف کوچک، بزرگ و اعداد استفاده شود، بهتر است رمز انتخاب شده شامل همه‌ی این موارد باشد.



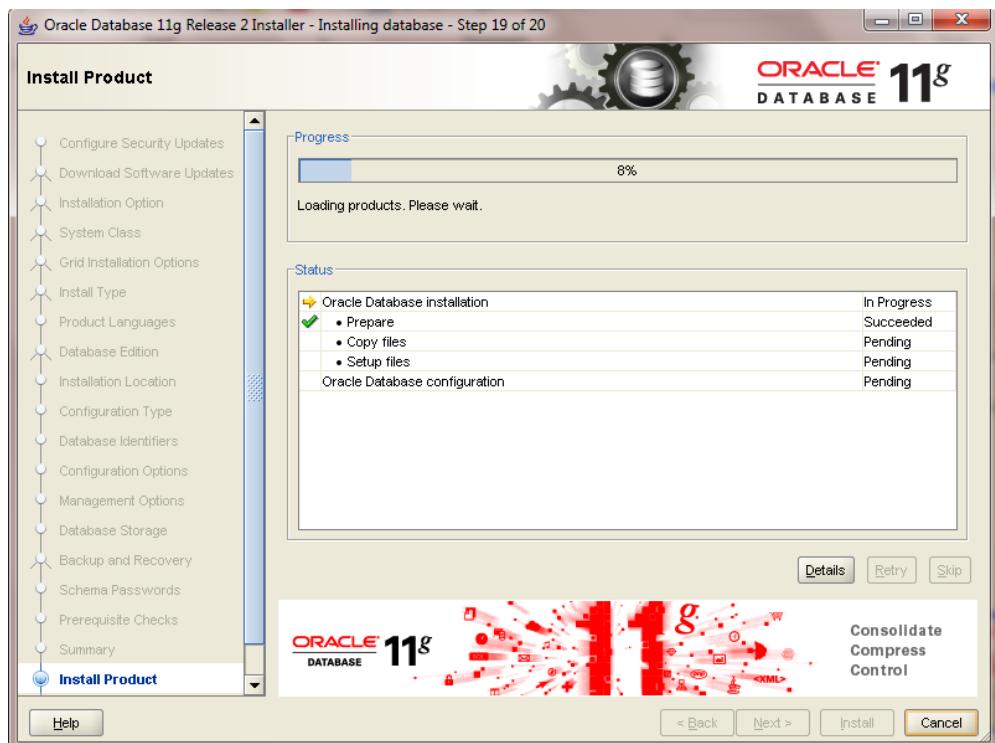
۱۵- پس از آن شروع به چک کردن موارد مشخص شده می‌کند، این صفحه به صورت زیر است:

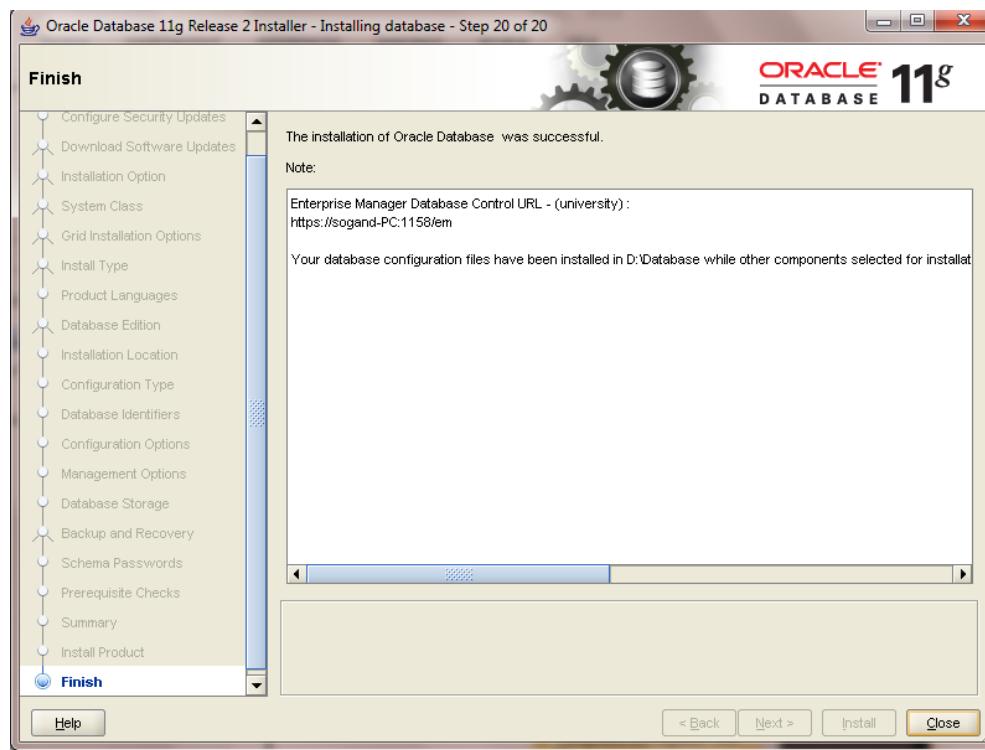
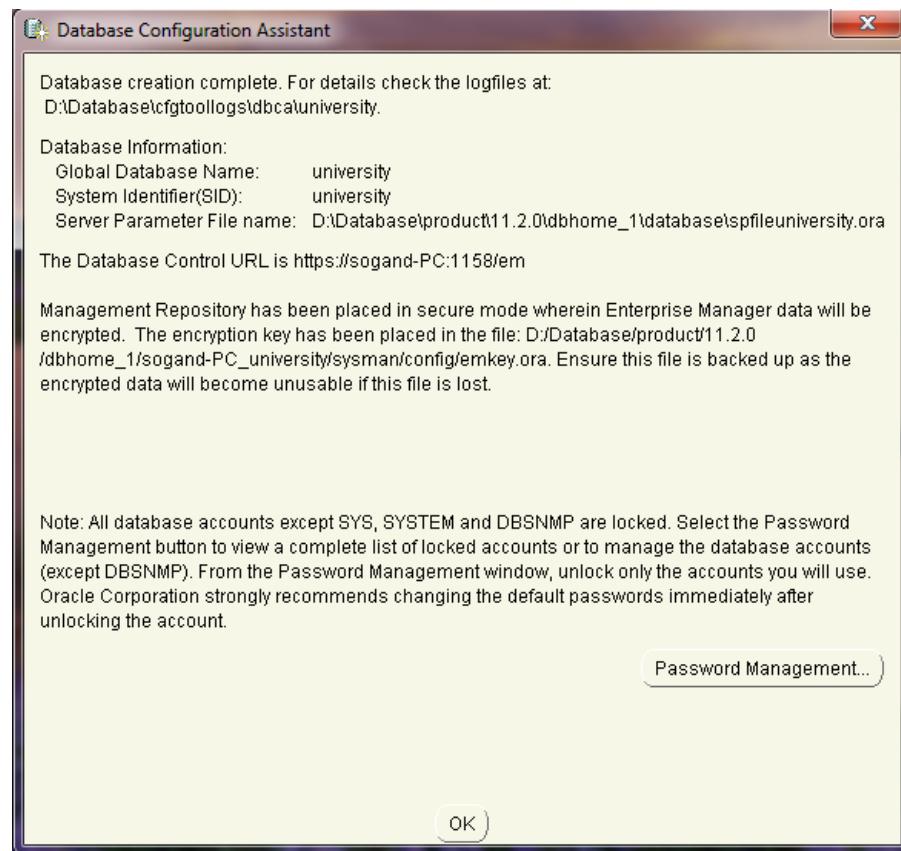


۱۶- در صورتی که با خطای روبرو شد، یا فشردن گزینه‌ی ignore all می‌توان به نصب ادامه داد. پس از آن با صفحه‌ای مطابق صفحه‌ی زیر روبرو می‌شویم:

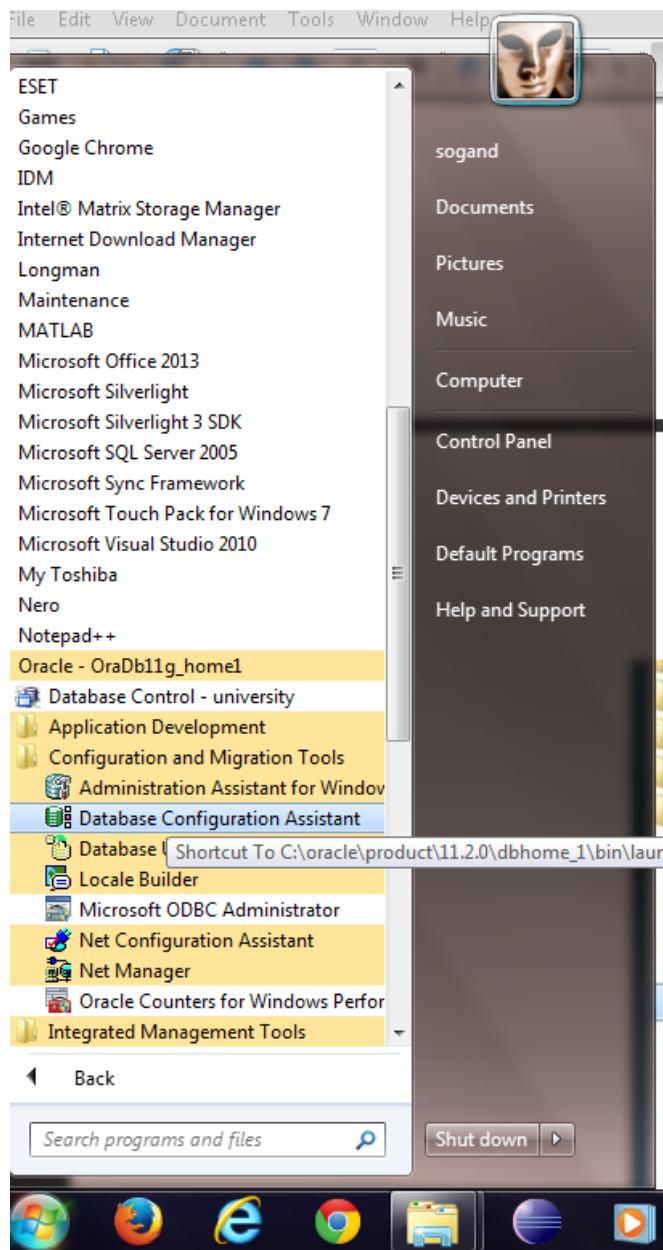


۱۷- پس از فشردن دکمه‌ی **install**، شروع به **install** کردن دیتابیس کرده و در نهایت با پیغام موفقیت نصب روبه رو می‌شویم:



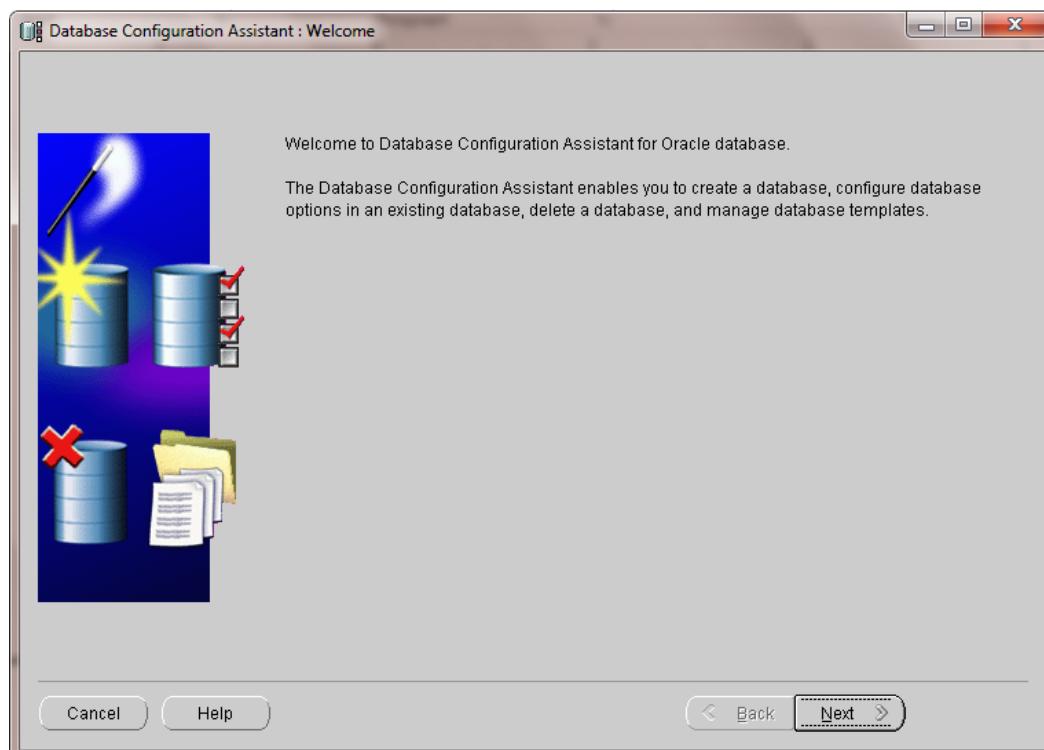


در ادامه با استفاده از Database Configuration Assistant به پیکربندی دیتابیس می‌پردازیم، Assistant را از آدرس زیر پیدا کرده و اجرا می‌کنیم:

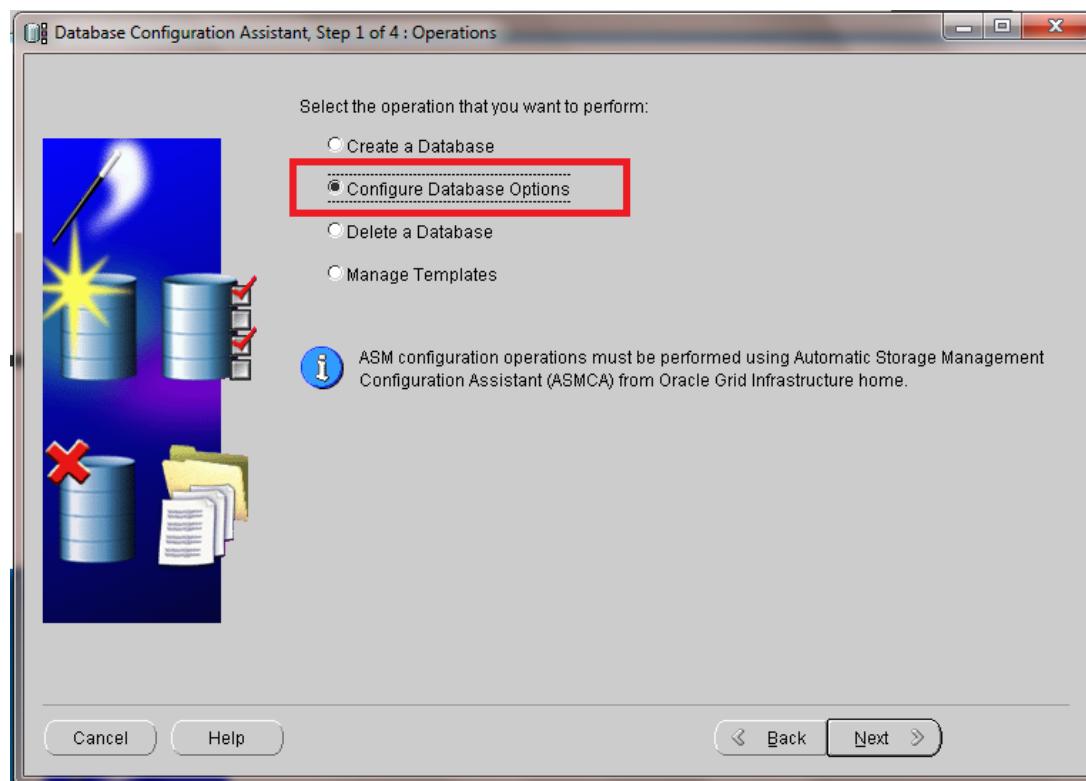


پس از آن مراحل پیکربندی به صورت زیر انجام می‌گیرد، در ادامه مراحل مهم با استفاده از شکل توضیح داده شده است:

۱- پس از اجرای این برنامه صفحه‌ی زیر باز می‌گردد، با فشردن دکمه‌ی next مراحل بعدی اجرا می‌شوند:

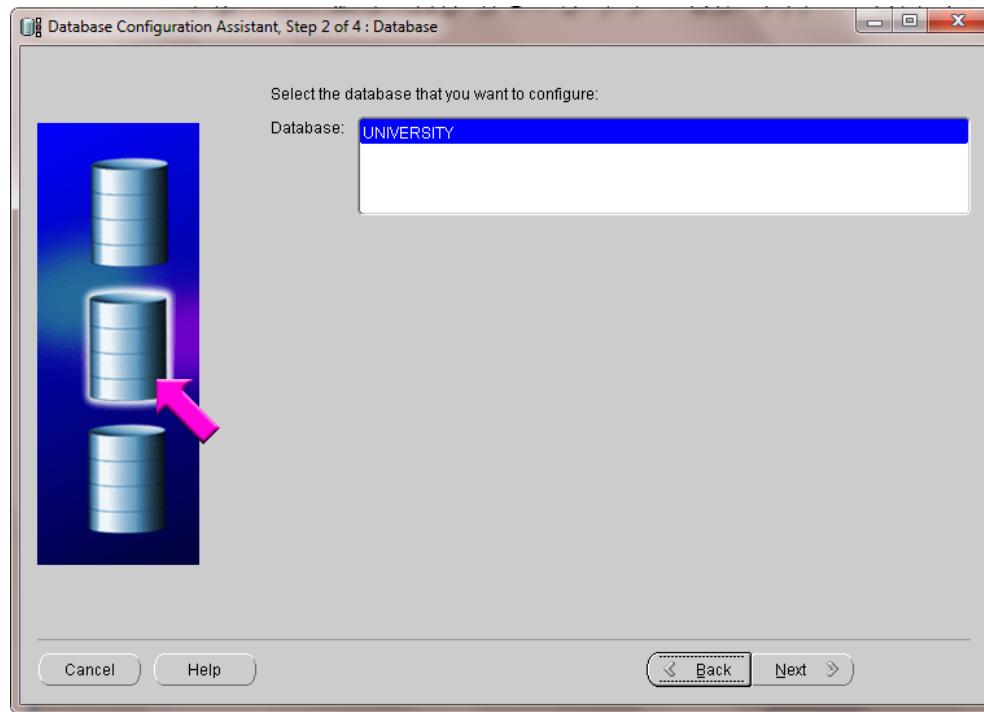


-۲

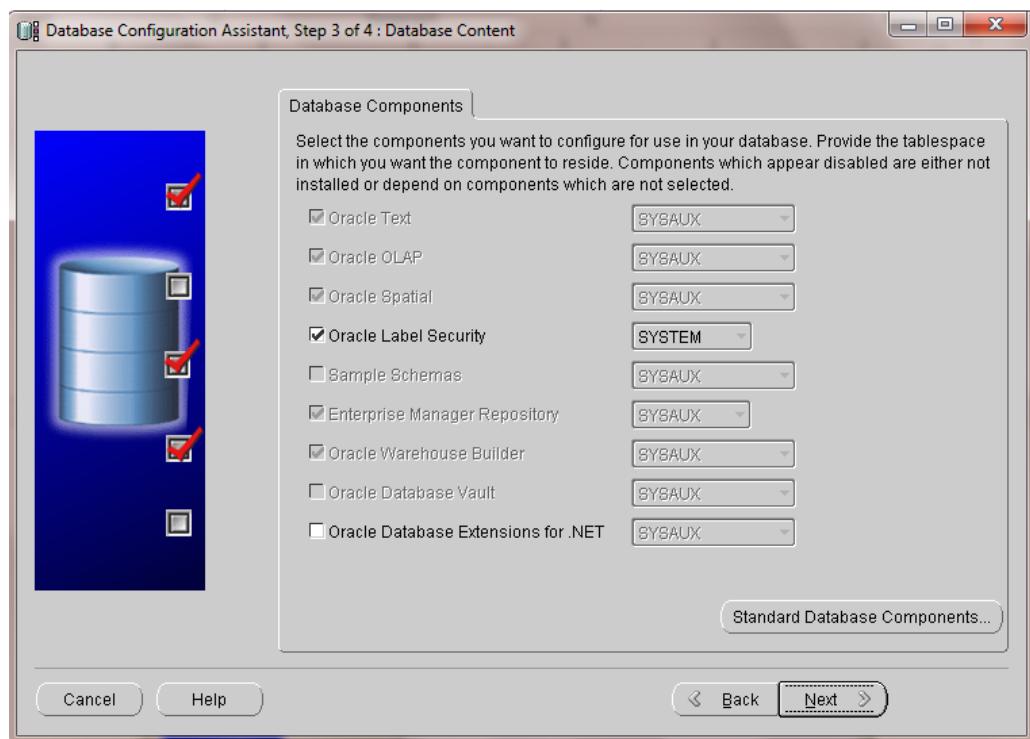


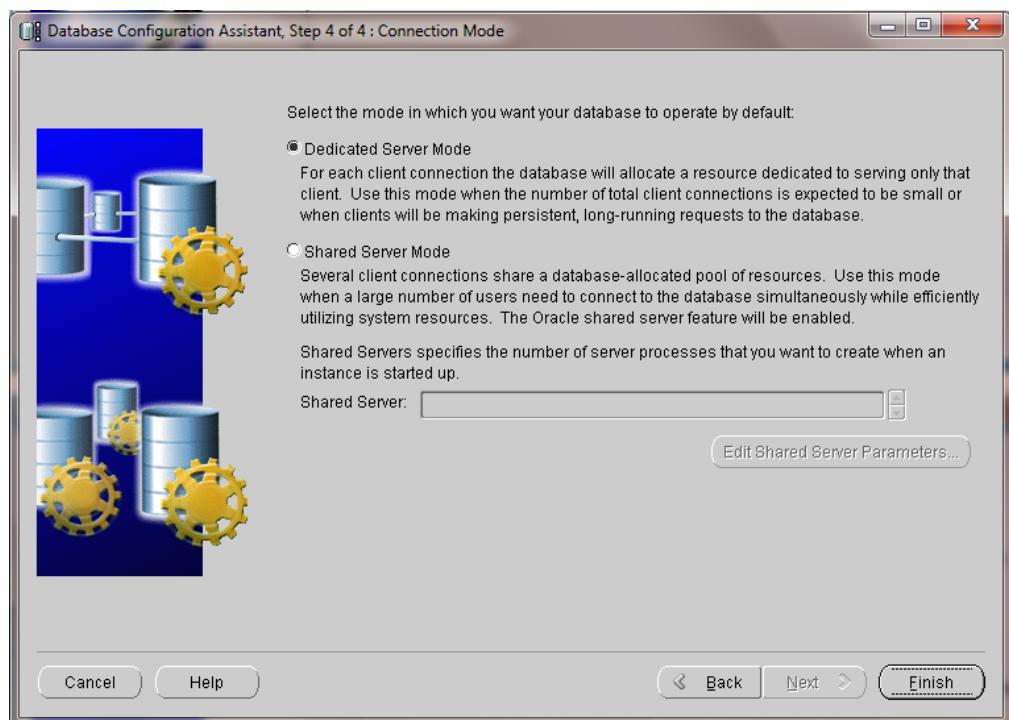
۱۲

-۳ در این مرحله دیتابیسی که قرار است پیکربندی برای آن انجام شود، انتخاب می‌گردد:



-۴





۶- با فشردن دکمه finish تغییرات بر روی دیتابیس مشخص شده، اعمال می‌گردد.

