



Gerenciamento de Dados e Informação

Revisão ER / Relacional / SQL / PL



Fernando Fonseca
Ana Carolina
Robson Fidalgo

cin.ufpe.br



Minimundo

- Construir um diagrama ERE para modelar conceitualmente o seguinte minimundo

Um clube social deseja informatizar suas atividades utilizando um banco de dados contendo as seguintes entidades

- Sobre Sócios devem ser armazenados dados de matrícula (identifica cada sócio), nome, endereço e situação

- Situação - com relação à tesouraria (Ex.: EM DIA, INADIMPLENTE, SUSPENSO)



cin.ufpe.br

2



Minimundo

- Construir o diagrama ERE (Cont.)

- Sobre Esportes devem ser armazenados dados de código (identifica cada esporte), modalidade e horários

- Modalidade (Basquete, Futebol, Natação, ...)



cin.ufpe.br

3



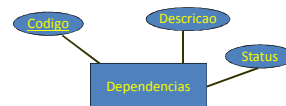
Minimundo

- Construir o diagrama ERE (Cont.)

- Sobre Dependências devem ser armazenados dados de código (identifica cada dependência), descrição e status

- Descrição (Quadra, Piscina, ...)

- Status (Livre, Interditada, Reservada,...)



cin.ufpe.br

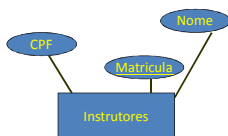
4



Minimundo

- Construir o diagrama ERE (Cont.)

- Sobre Instrutores devem ser armazenados dados de cpf, matrícula (identifica cada instrutor) e nome



cin.ufpe.br

5

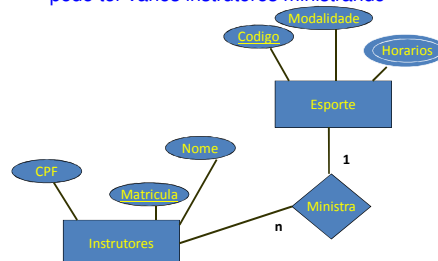


Minimundo

- Construir o diagrama ERE (Cont.)

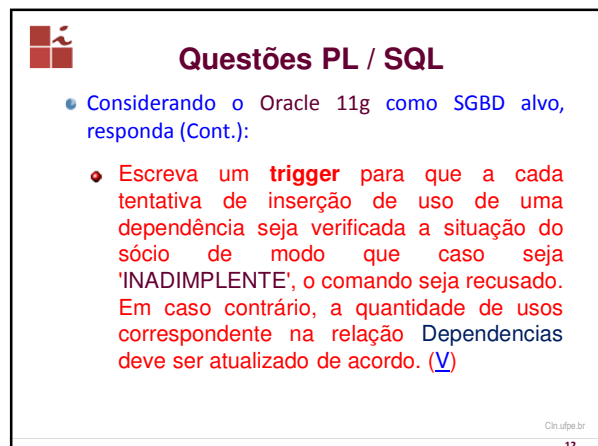
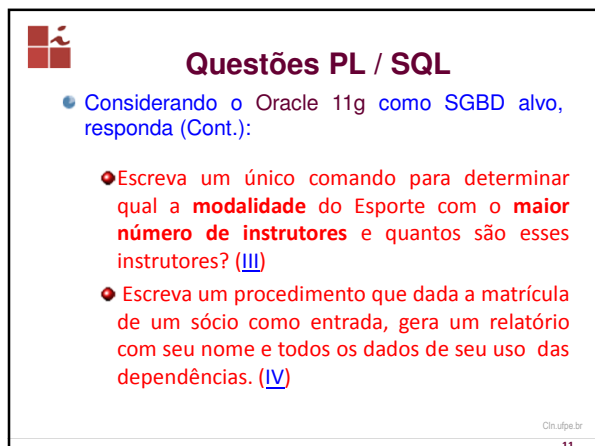
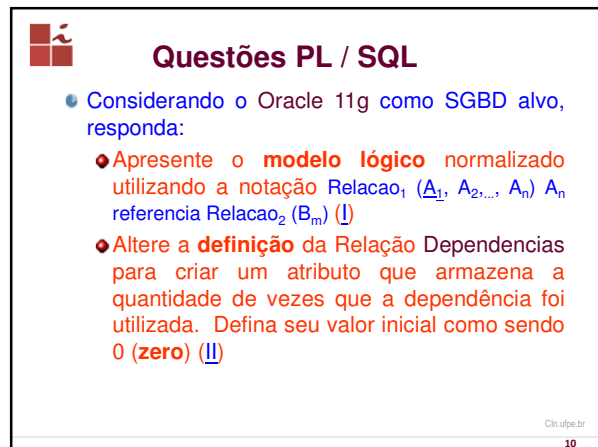
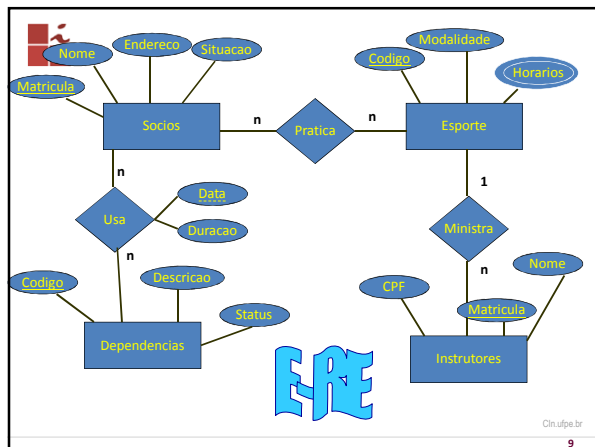
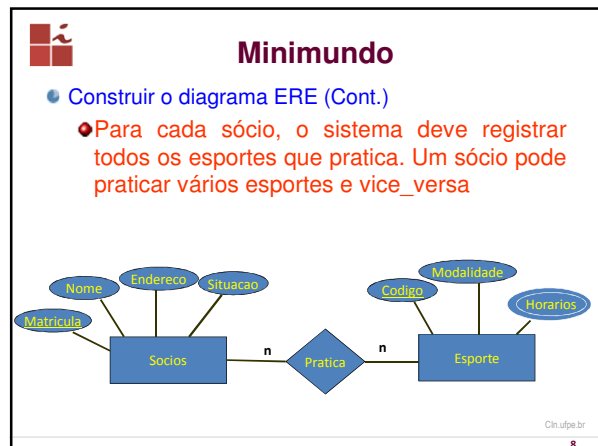
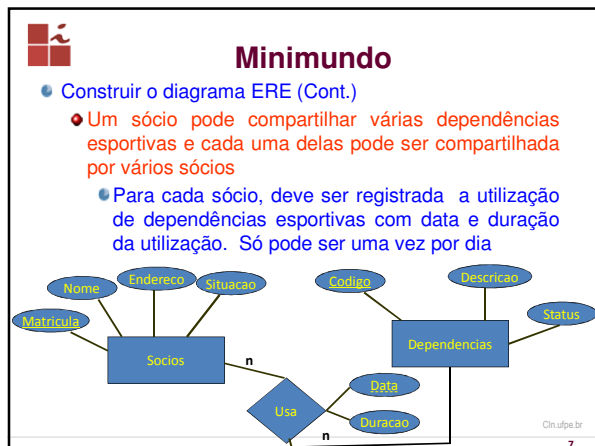
- Relacionamentos

- Cada instrutor ministra um único esporte, o qual pode ter vários instrutores ministrando



cin.ufpe.br

6





Questões PL / SQL

- Considerando o Oracle 11g como SGBD alvo, responda (Cont.):
 - Escreva uma função para retornar a descrição da dependência mais utilizada. (VI)
 - Alterar o trigger trig_3 criado, de modo que para cada inserção de uso de uma Quadra, seja alterada a quantidade de usos correspondente na relação Dependencias. (VII)
 - Escreva um trigger para duplicar o número de horas toda vez que for inserido um uso da dependência Campo (codigo = 4). (VIII)

CIn.ufpe.br

13



(I) Esquema Normalizado

- Notação pseudo-relacional

SOCIOS (Matricula, Nome, Endereco, Situacao)
 ESPORTE (Codigo, Modalidade)
 DEPENDENCIAS (Codigo, Descricao, Status)
 INSTRUTORES(CPF, Matricula, Nome, Cod_esporte)
 Cod_esporte referencia ESPORTE (Codigo)

CIn.ufpe.br

14



(I) Esquema Normalizado

- Notação pseudo-relacional (Cont.)

USA (CodigoDep, MatriculaS, Data, Duracao)
 CodigoDep referencia Dependencia (Codigo),
 MatriculaS referencia Socios (Matricula)
 PRATICA(MatS, CodE) MatS referencia Socio
 (Matricula), CodE referencia Esporte (Codigo)
 Horarios(CodE, HoraEsporte) CodE referencia Esporte
 (Codigo)

CIn.ufpe.br

15



Dados nas Relações

- Considerar que os seguintes dados foram inseridos nas relações do modelo

SOCIOS

Matricula	Nome	Endereco	Situacao
1	Maria Odete	Espinheiro	EM DIA
2	José cruz	Espinheiro	EM DIA
3	Márcio Maia	Boa Viagem	EM DIA
4	Beatriz Silva	Torre	INADIMPLENT

ESPORTE	Código	Modalidade	DEPENDENCIAS	Código	Descrição	Status
	1	Basquete		1	Quadra	Livre
	2	Natação		2	Piscina	Livre
	3	Futebol		3	Quadra	Livre
	4	Vôlei		4	Campo	Livre

CIn.ufpe.br

16



Dados nas Relações

- Considerando as relações com os seguintes dados (Cont.):

INSTRUTORES	CPF	Matricula	Nome	Cod_esporte
	12345687512	1	Sandro	1
	32145436789	2	Mara	2
	21345678613	3	Ivete	1
	54321734618	4	Marcos	3
	43276589715	5	Marina	4

USA	CodigoDep	MatriculaS	Data	Duracao
	1	1	10/01/2013	2
	3	2	09/01/2013	3
	3	1	20/02/2013	1

CIn.ufpe.br

17



Dados nas Relações

- Considerando as relações com os seguintes dados (Cont.):

PRATICA	<u>MatS</u>	<u>CodE</u>
	1	1
	2	2
	3	3
	4	2

HORARIOS	CodE	HoraEsporte
	3	10:00
	3	12:00
	3	15:00

CIn.ufpe.br

18



(II) Alteração de Tabela

- Acrescentar atributo a Dependencias

```
ALTER TABLE DEPENDENCIAS
ADD (Quant INTEGER DEFAULT 0);
```

DEPENDENCIAS

Codigo	Descricao	Status	Quant
1	Quadra	Livre	0
2	Piscina	Livre	0
3	Quadra	Livre	0
4	Campo	Livre	0

CIn.ufpe.br
19



(III) Consulta Modalidade

- Solução com um único comando SQL

```
(SELECT COUNT(*) as espor FROM
INSTRUTORES I, ESPORTE E where
I.cod_esporte = E.codigo
Group By E.codigo) T1
```

Determina a quantidade de Instrutores por modalidade de esportes

CIn.ufpe.br
20



(III) Consulta Modalidade

- Solução com um único comando SQL (Cont.)

```
(SELECT max(t1.espor) FROM
(SELECT COUNT(*) as espor FROM
INSTRUTORES I, ESPORTE E where
I.cod_esporte = E.codigo
Group By E.codigo) T1 )
```

Determina a quantidade de Instrutores por modalidade de esportes

Determina qual a maior quantidade de instrutores por modalidade de esporte

CIn.ufpe.br
21



(III) Consulta Modalidade

- Solução com um único comando SQL (Cont.)

```
(select codigo, count(*) as QT_Inst from Esporte
es, instrutores ins where ins.cod_esporte =
Es.codigo Group By Es.codigo having count(*) =
(SELECT max(t1.espor) FROM
(SELECT COUNT(*) as espor FROM
INSTRUTORES I, ESPORTE E where
I.cod_esporte = E.codigo
Group By E.codigo) T1 ) )T2
```

Determinar código da Modalidade esportiva com maior quantidade de instrutores

Determina a quantidade de Instrutores por modalidade de esportes

Determina qual a maior quantidade de instrutores por modalidade de esporte

CIn.ufpe.br
22



(III) Consulta Modalidade

- Solução com um único comando SQL (Cont.)

```
select sp. modalidade, T2. QT_Inst from Esporte sp,
(select codigo, count(*) as QT_Inst from Esporte
es, instrutores ins where ins.cod_esporte =
Es.codigo Group By Es.codigo having count(*) =
(SELECT max(t1.espor) FROM
(SELECT COUNT(*) as espor FROM
INSTRUTORES I, ESPORTE E where
I.cod_esporte = E.codigo
Group By E.codigo) T1 ) )T2
WHERE sp.codigo = t2.codigo;
```

Determinar modalidade com maior quantidade de instrutores

Determinar código da Modalidade esportiva com maior quantidade de instrutores

Determina a quantidade de Instrutores por modalidade de esportes

Determina qual a maior quantidade de instrutores por modalidade de esporte

CIn.ufpe.br
23



(III) Consulta Modalidade

- Solução com um único comando SQL (Cont.) - Resultado

```
select sp. modalidade, T2. QT_Inst from Esporte sp,
(select codigo, count(*) as QT_Inst from Esporte
es, instrutores ins where ins.cod_esporte =
Es.codigo Group By Es.codigo having count(*) =
(select max(t1.espor) from
(SELECT COUNT(*) as espor FROM
INSTRUTORES I, ESPORTE E where
I.cod_esporte = E.codigo
Group By E.codigo) T1 ) )T2
WHERE sp.codigo = t2.codigo;
```

MODALIDADE	QT_INST
Basquete	2

CIn.ufpe.br
24



(IV) Procedure

```
CREATE OR REPLACE PROCEDURE
relatorio (x socios.matricula%type) is
v_nome socios.nome%type;
v_desc dependencias.descricao%type; } Definir variáveis auxiliares
Cursor c_uso is
Select * from usa u where u.matriculaS = x; } Definir Cursor → Mais de um uso
v_usa c_uso%rowtype; } Definir variável do tipo do cursor
Begin
Select s.nome into v_nome from socios s where s.matricula = x; } Determina o sócio
Dbms_output.put_line('Sócio: ' || v_nome);
Dbms_output.put_line('Usos de Dependências');
Open c_uso; } Recupera informações de Uso
/
```

25



(IV) Procedure

```
Loop } Processa usos
Fetch c_uso into v_usa;
Exit when c_uso%NOTFOUND;
Select d.descricao into v_desc from dependencias d
where d.codigo = v_usa.cod_Dep;
Dbms_output.put_line('Dependência: ' || v_desc ||
'Data: ' || to_char(v_usa.data) || ' Duração: ' ||
to_char(v_usa.duracao));
End loop;
Close c_uso;
End relatorio;
/
```

Cln.ufpa.br

26



(IV) Procedure

Executando na interface SQL+

```
SQL> set serveroutput on;
SQL> create or replace procedure
relatorio (x socios.matricula%type) is
2 v_nome socios.nome%type;
3 v_desc dependencias.descricao%type;
4 Cursor c_uso is
5 Select * from usa u where u.matriculaS = x;
6 v_usa c_uso%rowtype;
7
8 Begin
9 Select s.nome into v_nome from socios s where s.matricula = x;
10 Dbms_output.put_line('Sócio: ' || v_nome);
11 Dbms_output.put_line('Usos de Dependências');
12 Open c_uso;
13 Loop
14 Fetch c_uso into v_usa;
15 Exit when c_uso%NOTFOUND;
16 Select d.descricao into v_desc from dependencias d
17 where d.codigo = v_usa.cod_Dep;
18 Dbms_output.put_line('Dependência: ' || v_desc ||
19 'Data: ' || to_char(v_usa.data) || ' Duração: ' ||
20 to_char(v_usa.duracao));
21 End loop;
22 Close c_uso;
23 End relatorio;
/
Procedure created.
SQL> exec relatorio(1);
Sócio: Maria da Silva
Usos de Dependências
Dependência: Quadra Data: 20-08-13 Duração: 2
Dependência: Quadra Data: 20-08-13 Duração: 1
SQL>
```

27



(V) Trigger

```
CREATE OR REPLACE TRIGGER trig_3
AFTER INSERT ON USA
FOR EACH ROW } Para permitir acesso aos valores manipulados
DECLARE
v_sit socios.situacao%type; } Variável auxiliar
BEGIN
SELECT s.situacao into v_sit FROM Socios s
where s.matricula = :NEW.matriculaS; } Determina situação do sócio
/
```

Cln.ufpa.br

28



(V) Trigger

```
IF v_sit = 'INADIMPLENTE' THEN
RAISE_APPLICATION_ERROR(-20011,
'Sócio em Atraso'); } Verifica situação do sócio
ELSE
UPDATE DEPENDENCIAS
SET QUANT = QUANT + 1
WHERE CODIGO = :NEW.COD_DEP; } Sócio adimplente
END IF;
END;
/
```

Cln.ufpa.br

29



(V) Trigger

- Teste 1 – Inserir o uso de piscina (Codigo = 2) para o sócio José Cruz (Matricula = 2) em 02/01/2013 por 3 h

Socios	2	José cruz	Espinheiro	EM DIA

Dependencias	2	Piscina	Livre	0

```
INSERT INTO usa VALUES(2, 2, TO_DATE('02/01/2013', 'dd/mm/yyyy'),
3);
```

Cln.ufpa.br

30

(V) Trigger

Teste 1 (Cont.)

SELECT * FROM Usa;

COD_DEP	MATRICULAS	DATA	DURACAO
1	1	10-JAN-13	2
3	1	20-FEB-13	1
2	2	02-JAN-13	3

SELECT * FROM Dependencias;

CODIGO	DESCRICAO	STATUS	QUANT
1	Quadra	Livre	0
2	Piscina	Livre	1
3	Quadra	Livre	0
4	Campo	Livre	0

Clin.ufpe.br 31

(V) Trigger

Teste 2

◆ Inserir o uso de quadra (Codigo = 1) para a sócia Beatriz (Matricula = 4) em 05/10/2015 por 2 h

Socios	4	Beatriz Silva	Torre	INADIMPLENTE
--------	---	---------------	-------	--------------

INSERT INTO usa VALUES(1, 4, TO_DATE('05/10/2015','dd/mm/yyyy'), 2);

Clin.ufpe.br 32

(V) Trigger

Teste 2 (Cont.)

Clin.ufpe.br 33

(VI) Function

```
CREATE OR REPLACE FUNCTION
MaisUtilizada return varchar2 is
v_desc dependencias.descricao%type; } Variável auxiliar
Begin
Select d.descricao into v_desc from dependencias d
where d.quant = (select max(quant) from dependencias);
Return v_desc; } Retorna valor calculado
End;
/
```

Determina a dependência mais utilizada

Teste 3

SELECT MaisUtilizada FROM Dual;

Clin.ufpe.br 34

(VI) Function

Executando na interface SQL+

Clin.ufpe.br 35

(VII) Trigger Alterado

```
CREATE OR REPLACE TRIGGER trig_3
AFTER INSERT ON USA
FOR EACH ROW } Para permitir acesso aos valores manipulados
DECLARE
v_sit socios.situacao%type; } Variável auxiliar
BEGIN
SELECT s.situacao into v_sit FROM Socios s
where s.matricula = :NEW.matriculaS;
IF v_sit = 'INADIMPLENTE' THEN
RAISE_APPLICATION_ERROR(-20011,
'Sócio em Atraso');
}
```

Determina situação do sócio

Clin.ufpe.br 36



(VII) Trigger Alterado

```
ELSE
  UPDATE DEPENDENCIAS
  SET QUANT = QUANT + 1
  WHERE CODIGO = :new.COD_DEP
  and Descricao = 'Quadra';
END IF;
END;
/
```

Sócio adimplente

Condição

Cin.ufpe.br

37



(VII) Trigger Alterado

- Considerar as tabelas USA e DEPENDENCIAS como mostradas abaixo

USA	COD_DEP	MATRICULAS	DATA	DURACAO
1	1	10-JAN-13	2	
3	2	09-JAN-13	3	
3	1	20-FEB-13	1	
2	2	02-JAN-13	3	

DEPENDENCIAS	CODIGO	DESCRICA	STATUS	QUANT
1	Quadra	Livre	1	
2	Piscina	Livre	0	
3	Quadra	Livre	0	
4	Campo	Livre	0	

38



(VII) Trigger Alterado

- Teste 4 – Inserir o uso de campo (Codigo = 4) para a sócia Maria Odete (Matrícula = 1) em 01/10/2015 por 5 h

Socios : 1 Maria Odete Espinheiro EM DIA

```
INSERT INTO usa VALUES(4, 1, TO_DATE('01/10/2015','dd/mm/yyyy'), 5);
```

SELECT * FROM usa;

Cin.ufpe.br

39



(VII) Trigger Alterado

USA	COD_DEP	MATRICULAS	DATA	DURACAO
1	1	10-JAN-13	2	
3	2	09-JAN-13	3	
3	1	20-FEB-13	1	
2	2	02-JAN-13	3	
4	1	01-OCT-15	5	

SELECT * FROM dependencias;

DEPENDENCIAS	CODIGO	DESCRICA	STATUS	QUANT
1	Quadra	Livre	1	
2	Piscina	Livre	0	
3	Quadra	Livre	0	
4	Campo	Livre	0	

40



(VII) Trigger Alterado

- Teste 5 – Inserir o uso de quadra (Codigo = 1) para o sócio Márcio Maia (Matrícula = 3) em 15/10/2015 por 4 h

Socios : 3 Márcio Maia Boa Viagem EM DIA

```
INSERT INTO usa VALUES(1, 3, TO_DATE('15/10/2015','dd/mm/yyyy'), 4);
```

Cin.ufpe.br

41



(VII) Trigger Alterado

SELECT * FROM usa;

USA	COD_DEP	MATRICULAS	DATA	DURACAO
1	1	10-JAN-13	2	
3	2	09-JAN-13	3	
3	1	20-FEB-13	1	
2	2	02-JAN-13	3	
4	1	01-OCT-15	5	
1	3	15-OCT-15	4	

SELECT * FROM dependencias;

DEPENDENCIAS	CODIGO	DESCRICA	STATUS	QUANT
1	Quadra	Livre	2	
2	Piscina	Livre	0	
3	Quadra	Livre	0	
4	Campo	Livre	0	

42



(VIII) Novo Trigger

- Considerar a tabela USA com os dados mostrados abaixo

USA	COD_DEP	MATRICULAS	DATA	DURACAO
	1	1	10-JAN-13	2
	3	2	09-JAN-13	3
	3	1	20-FEB-13	1
	2	2	02-JAN-13	3
	4	1	01-OCT-15	5
	1	3	15-OCT-15	4

Alterar tabela que sofre ação do comando que dispara o trigger → Tabela Mutante

43



(VIII) Novo Trigger

- Criar um trigger comum – Solução errada

```
CREATE OR REPLACE TRIGGER duplica_horas
AFTER INSERT OR UPDATE ON Usa
FOR EACH ROW
BEGIN
  IF :new.cod_dep = 4 THEN
    UPDATE Usa u SET u.duracao = u.duracao * 2 WHERE
      u.cod_dep = :new.cod_dep AND u.matriculas = :new.matriculas
      AND u.data = :new.data;
  END IF;
END;
```

Duplica horas
no uso de
Quadra

44



(VIII) Novo Trigger

- Inserir tupla

```
INSERT INTO usa VALUES(4, 2, TO_DATE('20/10/2015','dd/mm/yyyy'),
2);
```

```
*
ERROR at line 1:
ORA-04091: table U_FDFD.USA is mutating, trigger/function may not see it
ORA-06512: at "U_FDFD.DUPLICA_HORAS", line 3
ORA-04088: error during execution of trigger 'U_FDFD.DUPLICA_HORAS'
```

Necessário resolver o problema de Tabela Mutante

45



(VIII) Novo Trigger

- Utilizando autonomous transaction

```
CREATE OR REPLACE TRIGGER duplica_horas
BEFORE INSERT ON Usa
FOR EACH ROW
DECLARE
  PRAGMA AUTONOMOUS_TRANSACTION;
BEGIN
  IF :new.cod_dep = 4 THEN
    :new.duracao := :new.duracao * 2;
  END IF;
  COMMIT;
END;
```

Para permitir
alterar valores
inseridos pelo
comando que
dispara o trigger

Transação
do trigger
autônoma

Altera
duração

Efetiva a
transação
do Trigger

46



(VIII) Novo Trigger

- Teste 6 – Inserir o uso de campo (Codigo = 4) para a sócia Maria Odete (Matrícula = 1) em 22/10/2015 por 1 h

```
INSERT INTO Usa VALUES (4, 1, to_date('22/10/2015','dd/mm/yyyy'),
1);
```

```
SELECT * FROM usa;
```

USA	COD_DEP	MATRICULAS	DATA	DURACAO
	1	1	10-JAN-13	2
	3	2	09-JAN-13	3
	3	1	20-FEB-13	1
	2	2	02-JAN-13	3
	4	1	01-OCT-15	5
	1	3	15-OCT-15	4
	4	1	22-OCT-15	2

47