

Suggestion to solution the 25th october 2011

Question 1

A) Null (Connolly/Begg, p.103) Represents a null value for an attribute that is currently unknown or is not applicable for this tuple. Its not a value an therefore $\neq 0$.

[at the moment unknown value / not applicable value] and
[avoid trash-data inserts into db / not a value compared like zero (0)]

B) Stored procedure (Connolly/Begg, p.222) Stored procedures is a small program that can take parameters and be executed by users. Ypu can use them to do operations against several tables.

[parameters can be used] and
[to simplify operations/operations agains several tables]

C) Primary Key (Connolly/Begg, p.101) The candidate key that is selected to identify tuples uniquely within the relation. It can not be NULL.

[Selected candidate key] and
[to identify each row uniquely]

D) Deadlock (Connolly/Begg, p.590) When two or more transactions are waiting for locks to be realeased that are held by each other.

[An ever lasting loop] and
[locks held by two transactions]

E) View (Connolly/Begg, p.105) Dynamic result of one or more relational operations operating on the base relation to produce another relation. A view is a virtual relation that does not necessarily exist in the database but can be produced upon request by a particular user at the time of request.

[change-independence -> applications]
[virtual table / based on relational operations] and [used as a security mechanism]

F) Transaction (Connolly/Begg, p.570) An action, or series of actions, carried out by a single user or application program, that reads or updates the contents of a database.

[betraktas som en helhet och kan inkludera många operationer] and
[kan bara som helhet lyckas fullt ut eller göras ogjord vid fel/egenskaper ACID]

Question 2

ppt.

Question 3 (se 2.4 in the literature)

1. Data storage, retrieval and update. (Absolut viktigast)
2. A user-accessible catalog (viktig)
3. Transaction support
4. Concurrency control services (ja)
5. Recovery services (ja)
6. Authorization services
7. Support for data communication (ja)
8. Integrity services
9. Services to promote data independence
10. Utility services

Important with argumentation for your selection of functionalities!

1. Music Portal Inc. store data on customers' orders in a simple database consisting of one table (see figure below). The company's owner uses the database to keep track of the customers' orders. Analyze the table and discuss the design in terms of *redundancy, inconsistency and normal forms/normalization*. Also give suggestions (in terms of design) on a better database solution in the form of a final E/R model in 3NF. To receive full point - the answer must be structured in a good way. (12p)

Orders

OrderNo	ArtNo	Part	Weight	Quantity	Price	SupplierID	Supplier	SuppliersStatus
110133	EL200	JBL K-130	4	10	4000	L2200	Music shop Inc.	5
110298	H4100	Drum stool, black	3	2	1500	L4403	Drum store Inc.	3
110133	EL300	JBL Treble horn	2	6	2900	L2200	Musi shop Inc.	5
110298	H4101	Drum stool, chrome	3	18	2000	L4403	Drum store Inc.	2
110499	G2200	Guitar, acoustic	2	3	9500	L8806	Guitar Builders Inc.	4
↓	↓	↓	↓	↓		↓		↓

Prerequisites:

- the table has a primary key {OrderNo, ArtNo}
- various articles can be delivered by different suppliers. (Music Portal Inc. has never more than one provider of a specific article)
- Each supplier has a certain status (depending on delivery assurance)
- the music products have an article number, a price, a certain weight and can be found in different types or versions

Inconsistency: Data about same thing, but the value in the data differ – it is not consistent. For instance we can have the same name spelled in two different ways. Its close related to redundancy, if we eliminate redundancy, we also eliminate inconsistency.

In the order relation, we have marked some inconsistency with yellow color.

Redundancy: Is when the same data exists on several different places. If we have redundancy, there is a big risk of receiving inconsistent data if we modify data in the long run. Redundancy is an example of bad database design.

In the order relation we have Drum inc in two rows, marked with green color. We have trouble immediately if we don't change the name in all the redundant rows...

Normlization: To design the database in a proper way we need to split the table Order into many. We need to do it to avoid redundancy and inconsistency. For example is it problematic to update and delete data that is stored in many places (redundancy). If we forget to update a row, we risk that we receive inconsistent data. This means that we can have many answers to a question, then it can be difficult to make good decisions. Without data you can trust is the database unusable.

When we use normal forms we mention functional dependency (for each a (determinant) we have exactly one value on b (dependent)). For instance is city dependent on postal code, there is therefore one value for city for each postal code. We say that b is functionally dependent on a.

1st normal form is about that we only should have one value in each field. A primary key should also be chosen. In the example we can discuss the part. We need to split this field into two; partname and partdescription.

(OrderNo, ArtNo, Partname, Partdescription, Weight, Quantity, Price, SupplierID, Supplier, Supplierstatus)

2a normal form builds upon the 1NF. All non-key attributes should be dependent on the whole primary key. We break out all that only depends on half the primary key. That means the supplier and parts data. Supplier data we break out from the article since it depends on the supplierID. (if we don't fix the supplier non-key attributes into an own table, maybe we can do this by thinking on 3rd normal form in the next step – if you change supplierID you need to change value on supplier and supplierstatus as well, that's the problem to be eliminated)

(OrderNo, ArtNo, Quantity)

(ArtNo, Partname, Partdescription, Weight, Price, SupplierID)

(SupplierID, Supplier, Supplierstatus)

3d normal form builds upon the 2nd normal form and eliminates transitive dependencies between non key attributes. We need to split the dependency between non-key attributes:

(SupplierID, Supplier)

(Supplier, Supplierstatus)

Solution in 3NF:

(OrderNo, ArtNo, Quantity)

(ArtNo, Partname, Partdescription, Weight, Price, SupplierID)

(SupplierID, Supplier)

(Supplier, Supplierstatus)

5. Between the relations Customer and Order is a relationship to establish referential integrity (notice CustomerID). Put the answer of the execution of each question in the empty tables below. Answer as precise you can and don't forget to write the attribute names. To receive full point - the answer must be structured in a good way. (12p)

Customer

CustomerID	Company name	City	telephone	Contact person
1001	Music Portal Inc.	Stockholm	08-163241	Nils Larsson
1002	Musicality	Halmstad	035-121243	Eva Larsson
1003	Music Portal Inc.	Gotenburg	031-785411	Nanna Anka

Order

OrderID	Date	Delivery	CustomerID	SellerID
110032	2011-03-20	Cash on Delivery	1001	BK1
110038	2011-03-20	Home Delivery	1002	AK3
110211	2011-05-29	Customer Pick-up	1003	CK1
101935	2010-12-18	Customer Pick-up	1001	BK1

- a) `SELECT [Company name], City, telephone FROM [Customer] ORDER BY [Company name] ;`
- b) `SELECT * FROM [Customer] WHERE [Company name] LIKE '%Inc' ORDER BY [Company name];`
- c) `SELECT [Date], OrderID, Delivery FROM [Order] WHERE CustomerID = (SELECT DISTINCT CustomerID FROM [Customer] WHERE telephone = '031-785411');`
- d) `SELECT [Company name], City, telephone, [Contact person],[Date] FROM [Order], [Customer] WHERE [Order].CustomerID=[Customer].CustomerID AND SellerID='AK3';`
- e) `UPDATE [Customer] SET CustomerID=1004 WHERE CustomerID='1001';`
- f) `DELETE FROM [Order] WHERE CustomerID=1002;`

A

Company name	City	telephone
Musicality	Halmstad	035-121243
Music Portal Inc.	Gotenburg	031-785411
Music Portal Inc.	Stockholm	08-163241

B

CustomerID	Company name	City	Telephone	[Contact person]
1001	Music Portal Inc.	Gotenburg	031-785411	Nanna Anka
1003	Music Portal Inc.	Stockholm	08-163241	Nils larsson

c)

Date	OrderID	Delivery
110211	2011-05-29	Customer Pick-up

D)

[Company name]	City	telephone	[Contact person]	[Date]
Musicality	Halmstad	035-121243	Eva Larsson	11-03-20

e) You cant do that, depending on referential integrity.

The UPDATE statement conflicted with the REFERENCE constraint "FK_Order(order)_Kund(Customer)". The conflict occurred in database "Jesper_material", table "dbo.Order(order)", column 'KundID'.
The statement has been terminated.

f) Its no problem to delete the tuple. Result: 1 row affected

Question 6

ppt