Functional Dependencies

Anomalies (3)

- Bad designed schemas can result in redundancies and anomalies
- For example, having everything (all attributes and its corresponding values) in a single table is terrible
- Objective: To come up with a proper schema given a single table

Update Anomaly

```
// Price of standard rooms is replicated and was wrongly entered in one of the replicas
```

type	price	name	tel	address
…	…	…	…	…
standard	75	Hotel A		
…	…			
standard	76	Hotel A		
…	…			

- Because of repetition, easy to make mistakes
- Especially when need to update the attribute of certain rows to a new value
- If not all affected rows are updated to new value, database becomes inconsistent

Deletion Anomaly

```
// If Hotel B stops offering junior suites, then their price disappears from the database
```

type	price	name	tel	address
superior	…	Hotel A	…	…
standard		Hotel A		
suite		Hotel A		
superior		Hotel A		
standard	…	Hotel B		
suite	250	Hotel B		
junior suite	200	Hotel B		

- If we delete a row that contains a unique value of an attribute, then all details inferred from that particular value is also lost

Insertion Anomaly

```
// No hotels offer executive rooms yet, we cannot store their price
```

type	price	name	tel	address
…	…	…	…	…
…	…	…	…	…
executive	175	–	–	–

- We cannot use dummy values (dangerous) and null values (because attribute is part of primary key)
- This is because when all information is in a single table, all attributes form the primary key

+ Redundancies

```
Every time a hotel name is mentioned, its telephone and address is repeated
```

The same information is repeated, possibly unnessarily

Objectives

(1) Define functional dependencies (FD) and corresponding table
(2) Manipulate FDs (closure and equivalence)
(3) Reason about FDs (Armstrong's Axioms)
(4) Remove redundancy in dependencies and in sets of FDs (minimal cover)

Definition

For a relational schema R, a functional dependency from a set S of attributes of R to a set T of attributes of R exists

if and only if

For every instance of R, if 2 tuples in R agree on the values of the attributes in S, then they agree on the values of the attributes in T

We write it as:

$S \rightarrow T$

Example:

$\{A, B\} \rightarrow \{C, D\}$

Functional dependencies are:		
Integrity Constraints	Example: For R = {type, price, name, tel, address}, with functional dependency {type} → {price}	

	`// A FD is an integrity constraint that could be checked with:`
	
	`CHECK ( NOT EXISTS (`
	`SELECT *`
	`FROM R r1, R r2`
	`WHERE r1.type = r2.type`
	`AND r1.price <> r2.price))`
		- *** Note that this check cannot be implemented in Postgres
		- Currently, Postgres does not allow this as it can only handle simple conditions
		- The check here ensures that there is no two rooms of the same type with different prices

Best implemented as **Primary Keys** or **Unique Constraints**	Example: For R1 = {type, price}, R2 = {type, name} with functional dependency {type} → {price}	

	`// A FD is an integrity constraint that could be checked with:`
	
	`// In R1:`
	`type VARCHAR(36) UNIQUE`
	`//or//`
	`type VARCHAR(36) PRIMARY KEY`
	
	`// + Add a foreign key constraint`
	`// In R2:`
	`type VARCHAR(36) REFERENCES R1(type)`
		- This eliminates duplicate rows
		- Helps with breaking of table into multiple tables

Keys:		
Superkeys	A set of attributes whose knowledge determines the value of the entire tuple (*** implies everything)	Can be too big, smaller sets of candidate keys are better
Candidate keys	A minimal (for inclusion) set of attributes whose knowledge determines the value of the entire tuple	- Note that **minimal** is not the same as **minimum**: - Minimal – partial order/bottom of the hierarchy (not necessarily comparable) - Minimum – order
	Example: - {firstname, lastname} is not smaller than {passport} for inclusion - {lastname} is smaller than {firstname, lastname} for inclusion	
	Example: For R = {type, price, name, tel, address}, with a set of FDs: F = { {type}→{price}, {name}→{address}. {address}→{tel}, {tel}→{name} }	

	`// The candidate keys are:`
	`{type, name},`
	`{type, address},`
	`{type, tel}`
		Unique

Primary keys	If there are several candidate keys, only one is chosen to be the primary key	

Types of FDs:			
For: $X \rightarrow Y$	**Trivial**	**Non-trivial**	**Completely Non-trivial**
	$Y \subseteq X$ (Y is a **subset** or equals to X)	$Y \not\subset X$ (Y is **not** a **subset** of X)	$Y \not\subset X$ and $Y \cap X = \varnothing$ (**empty** intersection)
(Diagram)			
Example:	{type, name}→{type}	{type}→{price}	
		{type, name}→{price, name} where $Y \cap X$ = {name}	

Armstrong's Axioms	- It is possible to infer new FDs or simplify a set of given FDs - There are 3 tools in the Armstrong's axioms to create redundancy or find more about a given FD - They are **sound** and **complete** (≠ correct, means these 3 axioms are sufficient to proof anything – although may not be easy) - Q: Find the non-trivial FDs which are not completely non-trivial	
	Let X, Y, Z be subsets of the relation scheme of a relation R	
Reflexivity	If $Y \subseteq X$, then $X \rightarrow Y$	- Simplifies - Forms a trivial FD
Example:	For R1 = {type, price, name, tel, address} with the set of FDs F = {{type}→{price}, {name}→{tel, address}}	** **Note how proofs are written!** 1. It is a fact that {type} ⊂ {type, name} 2. Therefore {type, name} → {type} by reflexivity with (1)
Augmentation	If $X \rightarrow Y$, then $X \cup Z \rightarrow Y \cup Z$	Add on both sides
Example:	(Table and set of FDs same as above)	1. We know that {type}→{price} 2. Therefore {type, name}→{price, name} by augmentation of (1) with {name}
Transitivity	If $X \rightarrow Y$ and $Y \rightarrow Z$, then $X \rightarrow Z$	
Example:	For R1 = {type, price, name, tel, address} with the set of FDs F = {{type}→{price}, {name}→{address}, {address}→{tel}, {tel}→{name}}	1. We know that {name}→{address} 2. We know that {address}→{tel} 3. Therefore {name}→{tel} by transitivity of (1) and (2)
Other axioms: Weak-Augmentation	If $X \rightarrow Y$, then $X \cup Z \rightarrow Y$	Proof: 1. Let R be a relation scheme 2. Let $X \rightarrow Y$ be a FD on R 3. Therefore $X \cup Z \rightarrow Y \cup Z$, by augmentation of (2) with {Z} 4. We know that $Y \cup Z \rightarrow Y$ by reflexivity because $Y \subset (Y \cup Z)$ 5. Therefore $X \cup Z \rightarrow Y$ by transitivity of (3) and (4)

Example: Finding keys

Set of Functional Dependencies

Closure of a **set of FDs**	- Notation for a closure of F, a set of FDs: F+ - F+ is the set of all FDs that F entails (it carries equivalent information as F) - F+ can be gotten by applying Armstrong's Axioms in all possible ways until nothing new is created (called a fixpoint)
Example	Consider the relation schema R = {A, B, C, D} with set of FDs, F = {{A}→{B}, {B, C}→{D}}

	F+ = {
	
	}

Equivalence	Two sets of functional dependencies F and G are equivalent $F \equiv G$ if and only if F+ = G+ (i.e. everything implies everything)

Set of Attributes

Attribute Closure ***	- For a set S of attributes, the closure of S (with respect to a set of FDs, F), S+, is the maximum set of attributes such that S → S+ (as a consequence of F) - Closure size is exponential, 2^N where N = # attr or $2^N - 1$ (if we do not consider the empty set) - Q: Given {A}, {B}, tell everything that they imply

Algorithm

Input	Output
R, a relation scheme F, a set of FDs $X \subset R$	X+, the closure of A wrt F

```
X(0)  := X
Repeat
        X(i+1)  := X(i)  ∪ A
        where A is the union of the sets Z of attributes
                    s.t. there exists Y → Z in F
                                and Y ⊂ X(i)
Until X(i+1)  := X(i)
Return X(i+1)
```

Example 1

For R = {A, B, C, D}, with set of FDs F = {{A}→{C}, {B}→{D}}	▪ {A}+ = {A, C} ▪ {B}+ = {B, D} ▪ {A, B}+ = {A, B, C, D} (Minimal – everything else is a superkey) ▪ etc (There are $2^N+1/2^N$ possible sets)

<table>
<tr>
<td>Example 2
(Using algorithm)</td>
<td>For
$R = \{A, B, C, D, E, G\}$,
with set of FDs
$F = \{\{A, B\}\rightarrow\{C\}, \{C\}\rightarrow\{A\}, \{B, C\}\rightarrow\{D\},$
$\{A, C, D\}\rightarrow\{B\}, \{D\}\rightarrow\{E, G\},$
$\{B, E\}\rightarrow\{C\}, \{C, G\}\rightarrow\{B, D\},$
$\{C, E\}\rightarrow\{A, G\}\}$
and
$X = \{B, D\}$
Q: Find X+</td>
<td>1) Calculate given set of attributes
2) Look at each FD, if LHS is something in LHS of current set $X^{(N)}$, then can add in RHS
(e.g. {<u>A, B</u>}→{<u>A, B</u>, C})
3) Always choose a FD that can introduce more new attributes
4) Once a FD is used, no need to use it again
5) Stop when nothing new is created (fixpoint) or when LHS is superkey
▪ $X^{(0)} = \{B, D\}$
▪ Since $\{D\}\rightarrow\{E, G\}$,
$X^{(1)} = \{B, D, E, G\}$
▪ Since $\{B, E\}\rightarrow\{C\}$,
$X^{(2)} = \{B, D, E, G, C\}$
▪ Since $\{C, E\}\rightarrow\{A, G\}$,
$X^{(3)} = X^{(4)}$
$= X+ = \{A, B, C, D, E, G\}$</td>
</tr>
<tr>
<td rowspan="2">Equivalence</td>
<td colspan="2">- Q: Testing equivalence based on attribute closures</td>
</tr>
<tr>
<td>Let R be a relational scheme
Let F, G be 2 sets of FDs on R
▪ For each $\{X \rightarrow Y\}$ in F
1) compute $X^{+(G)}$
(i.e. compute X+ based on FDs in G)
2) if $Y \not\subset X^{+(G)}$, return false
▪ For each $\{X \rightarrow Y\}$ in G
1) compute $X^{+(F)}$
2) if $Y \not\subset X^{+(F)}$, return false
▪ return true
(i.e. equivalent set of FDs)</td>
<td>Example:
If F contains {A}→{B, C}
but $\{A\}^{+(G)}\rightarrow\{A, B\}$

then {A}→{C} is entailed by F
but not by G

Therefore F and G are not equivalent</td>
</tr>
<tr>
<td></td>
<td colspan="2"></td>
</tr>
<tr>
<td>Minimal Set of Dependencies</td>
<td colspan="2">A set of dependencies F is minimal if and only if:
1) Every RHS is a single attribute
2) For no FD {X→A} in F and proper subset Z of X is F - {X→A} ∪ {Z→A}
(i.e. for 2 FDs with the same RHS, if one LHS is a subset of the other LHS, choose the FD with the smaller set)
3) For no FD {X→A} in F is the set F - {X→A} equivalent to F</td>
</tr>
<tr>
<td>Example</td>
<td colspan="2">For 1),
{A}→{B, C} ⇒ {A}→{B} and {A}→{C}
For 2),
{X}→{A}
{Y}→{A} is minimal since Y ⊆ X
Therefore no need to keep {X}→{A}
For 3),
{A}→{B}
{B}→{C}
{C}→{A}</td>
</tr>
</table>

Minimal Cover	- A set of FDs F is a minimal cover of a set of FDs G if and only if 1) F is minimal 2) F is equivalent to G - Every set of FDs has a minimal cover - There might be several **different minimal covers** of the **same set** - Q: Find the minimal cover of a set of FDs and reduce redundancy - A: Answers are **not** unique, depends on order of steps in algorithm - Challenge is to cover all cases, if there is redundancy left – if will be reflected in schema
*** Algorithm	- At every step, transform the set of FDs into an equivalent set - Steps (1), (2), (3) can be applied iteratively in various orders - Algorithm: (1) Simplify the **RHS** (by splitting into single attribute on RHS) - Find FDs with RHS with more than 1 attribute, **split** (2) Simplify the **LHS** (by finding smallest possible algo/eliminate redundant lists) - Find if there exists a **LHS** that is a **subset** of it in set of FDs (**same RHS**), use that FD instead - Or, use **Armstrong's Axioms** to find **redundant** FDs (finding attribute closure will make it faster in identifying) (3) Simplify the **entire set** - Find FDs which are redundant (i.e. can already be gotten by Armstrong's Axioms through other FDs)
Extended Minimal Cover	Add a fourth step to the algorithm: (4) Regroup (combine LHS, undo step 1) - Combine FDs with same LHS
Example	Set of FDs, F = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {A, C, D}→{B}, {D}→{E, G}, {B, E}→{C}, {C, G}→{B, D}, {C, E}→{A, G}}
(Example) Step (1) - Look for common RHS	F = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {A, C, D}→{B}, **{D}→{E, G}**, {B, E}→{C}, **{C, G}→{B, D}**, **{C, E}→{A, G}**} F' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {A, C, D}→{B}, ~~{D}→{E, G}~~, {D}→{E}, {D}→{G}, {B, E}→{C}, ~~{C, G}→{B, D}~~, {C, G}→{B}, {C, G}→{D}, ~~{C, E}→{A, G},~~ {C, E}→{A}, {C, E}→{G}}
(Example) Step (2) - Check if LHS is a subset and can be reduced	F' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {A, C, D}→{B}, {D}→{E}, {D}→{G}, {B, E}→{C}, {C, G}→{B}, {C, G}→{D}, {C, E}→{A}, {C, E}→{G}} Since {C} is a subset of {C, E} and {C}→{A}, then {C, E}→{A} is redundant. Since {D}→{G}, therefore {C, D}→{C, G} by augmentation of it with {C}. Since {C, G}→{B}, therefore {C, D}→{B} by transitivity. Therefore {A, C, D}→{B} is redundant and can be replaced by {C, D}→{B}. F'' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, ~~{A, C, D}→{B},~~ {C, D}→{B}, {D}→{E}, {D}→{G}, {B, E}→{C}, {C, G}→{B}, {C, G}→{D}, ~~{C, E}→{A},~~ {C, E}→{G}}

(Example) Step (3) - Check if LHS implies something new and this something new is a subset of LHS (trivial/cancel)	F' ' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {C, D}→{B}, {D}→{E}, {D}→{G}, {B, E}→{C}, {C, G}→{B}, {C, G}→{D}, {C, E}→{G}} Since {C, G}→{D}, therefore {C, G}→{C, D} by augmentation of it with {C}. Since {C, D}→{B}, therefore {C, G}→{B} by transitivity. Therefore {C, G}→{B} is redundant. F' ' ' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {C, D}→{B}, {D}→{E}, {D}→{G}, {B, E}→{C}, ~~{C, G}→{B}~~, {C, G}→{D}, {C, E}→{G}} F' ' ' is the minimal cover.
(Example) Step (4) - Obtain **extended** minimal cover by undoing Step (1) on minimal cover	F' ' ' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {C, D}→{B}, {D}→{E}, {D}→{G}, {B, E}→{C}, {C, G}→{D}, {C, E}→{G}} F' ' ' ' = {{A, B}→{C}, {C}→{A}, {B, C}→{D}, {C, D}→{B}, ~~{D}→{E}, {D}→{G},~~ {D}→{E, G} {B, E}→{C}, {C, G}→{D}, {C, E}→{G}} F' ' ' ' is the extended minimal cover.