Tutorial: Dependencies, entity-relationship modelling and the Chase

1. Consider the entity-relationship diagram of Figure 1.

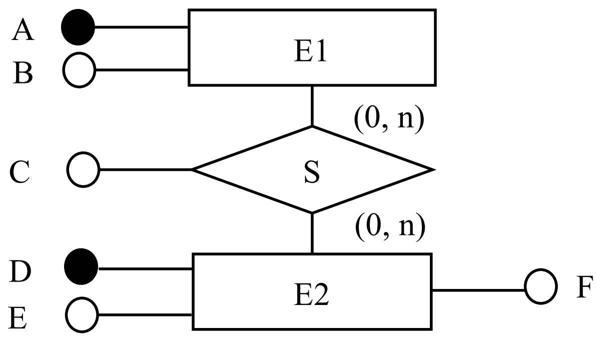

Figure 1: Entity-relationship diagram.

   (a) Without other knowledge than that captured by the entity-relationship diagram, what are the functional and multivalued dependencies?

   Solution: The entity-relationship diagram tells us the following functional dependencies.

   $$\Sigma = \{\{A\} \rightarrow \{B\}, \{D\} \rightarrow \{E, F\}, \{A, D\} \rightarrow \{C\}\}$$

   Multivalued dependencies depend on the translation of this design into tables. The entity-relationship diagram and its translation will only result in "not interesting" multi-valued dependencies, that is dependencies that are trivial or correspond to the table or the functional dependencies. If we canonically translate this design into three tables, we have $\{A\} \twoheadrightarrow \{B\}$ and $\{D\} \twoheadrightarrow \{E, F\}$ and many more (none of them being useful for normalisation), for instance.

   Note that additional knowledge of the application could tell us additional functional and multivalued dependencies not captured by the design.

   For instance, we could know that $\{E\} \rightarrow \{F\}$ (which would suggest that the entity-relationship design is probably missing entities and relationships that have been merged too early), which we could use to produce a design in the Boyce-Codd normal by splitting the table $R_3(D, E, F)$ into two tables $R_{3.2}(\underline{D}, E)$ and $R_{3.2}(\underline{E}, F)$.

   For instance, we could know that $\{E\} \twoheadrightarrow \{F\}$, which we could use to produce a design in the fifth normal by splitting the table $R_3(D, E, F)$ into two tables $R_{3.2}(\underline{D}, E)$ and $R_{3.2}(\underline{E, F})$.

2. Consider the relational schema $R = \{A, B, C, D, E\}$ with the following set of functional and multi-valued dependencies.

   $\Sigma = \{\{C\} \rightarrow \{A\}, \{D\} \rightarrow \{D, B\}, \{B\} \rightarrow \{E\}, \{E\} \twoheadrightarrow \{A, D\}, \{A, B, D\} \rightarrow \{A, B, C, D\}, \{B\} \rightarrow \{D\}\}$

   (a) Prove that $\{E\} \rightarrow \{D\}$ using the Armstrong and multi-valued dependencies axioms.

   Solution:

   1. We know that $\{E\} \twoheadrightarrow \{A, D\}$.
   2. We know that $\{B\} \rightarrow \{D\}$.
   3. We see that $\{D\} \subset \{A, D\}$.
   4. We see that $\{B\} \cap \{A, D\} = \emptyset$.

5. Therefore $\{E\} \to \{D\}$ by Coalescence of (1), (2), (3) and (4).

   Q.E.D.

Try the same question with the Chase (answer not provided).

3. Consider the relation $R(A, B, C, D, E, G)$ with the following set, F, of functional and multi-valued dependencies.

   $F = \{\{A, B\} \to \{C\}, \{A, B\} \twoheadrightarrow \{E\}, \{C, D\} \twoheadrightarrow \{A, B\}\}$

   (a) Prove that the decomposition of R into $R_1(A, B, C, D, G)$ and $R_2(C, D, E)$ is lossless using the Chase algorithm (as shown in the lecture).

Solution:

1. Initial table.

A	**B**	**C**	**D**	**E**	**G**
a1	b1	c1	d1	e1	g1
a2	b2	c2	d2	e2	g2

2. We want to chase $\{C, D\} \twoheadrightarrow \{E\}$, make C and D values the same.

A	**B**	**C**	**D**	**E**	**G**
a1	b1	**c1**	**d1**	e1	g1
a2	b2	**c1**	**d1**	e2	g2

3. Apply $\{C, D\} \twoheadrightarrow \{A, B\}$ by copying two t-uples that have the same C and D values but swapping their A and B values.

A	**B**	**C**	**D**	**E**	**G**
a1	b1	c1	d1	e1	g1
a2	b2	c1	d1	e2	g2
a2	**b2**	c1	d1	e1	g1
a1	**b1**	c1	d1	e2	g2

4. Apply $\{A, B\} \twoheadrightarrow \{E\}$ by copying two t-uples by copying two t-uples that have the same A and B values but swapping their A and E values.

A	**B**	**C**	**D**	**E**	**G**
a1	b1	c1	d1	e1	g1
a2	b2	c1	d1	e2	g2
a2	b2	c1	d1	e1	g1
a1	b1	c1	d1	e2	g2
a1	b1	c1	d1	**e2**	g1
a2	b2	c1	d1	**e1**	g2
a2	b2	c1	d1	**e2**	g1
a1	b1	c1	d1	**e1**	g2

5. Applying $\{A, B\} \to C$ do not change the table.
6. Sort the table and proved that $\{C, D\} \twoheadrightarrow \{E\}$

A	**B**	**C**	**D**	**E**	**G**
a1	b1	c1	d1	e1	g1
a1	b1	c1	d1	e2	g1
a2	b2	c1	d1	e1	g2
a2	b2	c1	d1	e2	g2
a2	b2	c1	d1	e1	g1
a2	b2	c1	d1	e2	g1
a1	b1	c1	d1	e2	g2
a1	b1	c1	d1	e1	g2

4. Consider the relation $R(A, B, C, D, E, F, G)$ with the following set, Σ, of functional dependencies.

   $\Sigma = \{\{A, B\} \rightarrow \{C\}, \{C\} \rightarrow \{D, E\}, \{E\} \rightarrow \{D\}, \{F\} \rightarrow \{G\}\}$

   (a) Prove that the decomposition of R into $R_1(A, B, C, D, E)$ and $R_2(A, B, F, G)$ is lossless using the Chase algorithm (as shown in the lecture).

Solution:

1. Initial table.

A	**B**	**C**	**D**	**E**	**F**	**G**
a1	b1	c1	d1	e1	f1	g1
a2	b2	c2	d2	e2	f2	g2

2. We want to chase $\{A, B\} \twoheadrightarrow \{C, D, E\}$, make A and B values the same.

A	**B**	**C**	**D**	**E**	**F**	**G**
a1	**b1**	c1	d1	e1	f1	g1
a1	**b1**	c2	d2	e2	f2	g2

3. Apply $\{A, B\} \rightarrow \{C\}$, make C with the same A and B values the same.

A	**B**	**C**	**D**	**E**	**F**	**G**
a1	b1	**c1**	d1	e1	f1	g1
a1	b1	**c1**	d2	e2	f2	g2

4. Apply $\{C\} \rightarrow \{D, E\}$, make D and E with the same C values the same.

A	**B**	**C**	**D**	**E**	**F**	**G**
a1	b1	c1	**d1**	**e1**	f1	g1
a1	b1	c1	**d1**	**e1**	f2	g2

5. Applying $\{E\} \rightarrow \{D\}$ does not change the table.
6. Applying $\{F\} \rightarrow \{G\}$ does not change the table.
7. Proved that $\{A, B\} \twoheadrightarrow \{C, D, E\}$.

5. Consider the relation $R(A, B, C, D, E)$ with the following set, Σ, of functional dependencies.

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

   (a) Check whether the decomposition of R into $R_1(A, E)$, $R_2(C, D)$ and $R_3(A, B, C)$ is lossless or lossy using the Chase algorithm with the distinguished attributes.

Solution:

1. Initial table.

	A	**B**	**C**	**D**	**E**
R_1	a				a
R_2			a	a	
R_3	a	a	a		

2. Apply $\{A\} \rightarrow \{B, C\}$

	A	**B**	**C**	**D**	**E**
R_1	a	**a**	**a**		a
R_2			a	a	
R_3	a	a	a		

3. Applying $\{B\} \rightarrow \{A\}$ does not change the table.

4. Apply $\{C\} \rightarrow \{D\}$

	A	**B**	**C**	**D**	**E**
R_1	a	a	a	**a**	a
R_2			a	a	
R_3	a	a	a	**a**	

5. R_1 has distinguished variables in all of the columns, therefore the decomposition is lossless.