**Relational Model:** store all data in relations

|  |  |
| --- | --- |
| Relation | Table |
| Attribute | Column in a table |
| Record /Tuple | Row in a table |

**Entity-Relationship (ER) Diagrams**:

A pictorial and intuitive way for translating the conceptual model into a set of tables

|  |  |
| --- | --- |
| Screen Clipping | **Entity Set / Object Class**  Entity = Real-world object (e.g. a bar)  Entity Set = Collection of similar objects (e.g. a set of bars) |
| Screen Clipping | **Attribute**  Property of an entity set |
|  | **Relationship**  Connection between 2 entity sets  E.g. Persons buy products |

|  |  |  |
| --- | --- | --- |
| Screen Clipping | Screen Clipping | Screen Clipping |

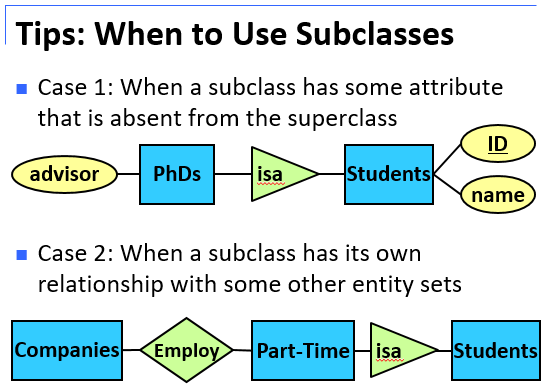
|  |  |
| --- | --- |
| Screen Clipping | Screen Clipping |

**Some constraints that the entity sets and relationships should satisfy:**

* Key (underlined attributes)
* Referential Integrity
* Degree constraint

|  |  |
| --- | --- |
| Screen Clipping | Screen Clipping |

|  |  |
| --- | --- |
| Screen Clipping | Screen Clipping |

**Design Principle:**

1. Be Faithful
2. Avoid Redundancy
3. Keep It Simple
4. Don’t over-use Weak Entity Sets

|  |  |
| --- | --- |
| Screen Clipping | Screen Clipping |
| Screen Clipping | Screen Clipping |
|  |  |

**Subclass -> Relation**

|  |  |  |
| --- | --- | --- |
| **Screen Clipping** | **Screen Clipping** | **Screen Clipping** |

|  |  |
| --- | --- |
| **Screen Clipping** | **Screen Clipping** |

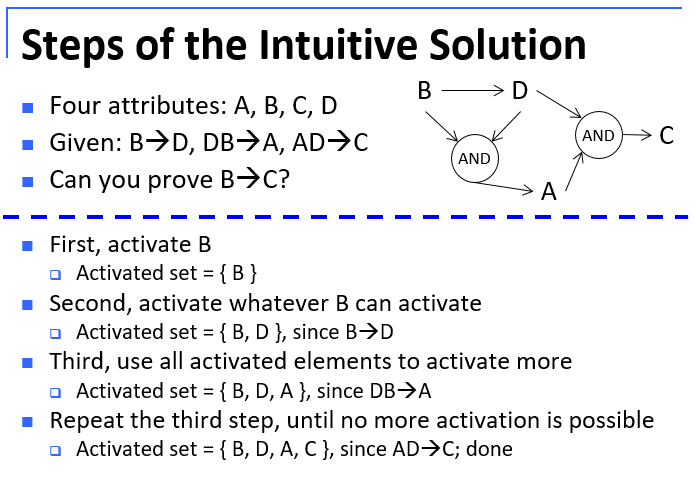
**Functional Dependencies:** correlations among attributes a table

|  |
| --- |
| NRIC 🡪 Name, but not Name 🡪 NRIC  \*NRIC decides Name, but not vice versa  **Meaning**: There do not exist two persons that have the same NRIC but different Name.  Screen Clipping |

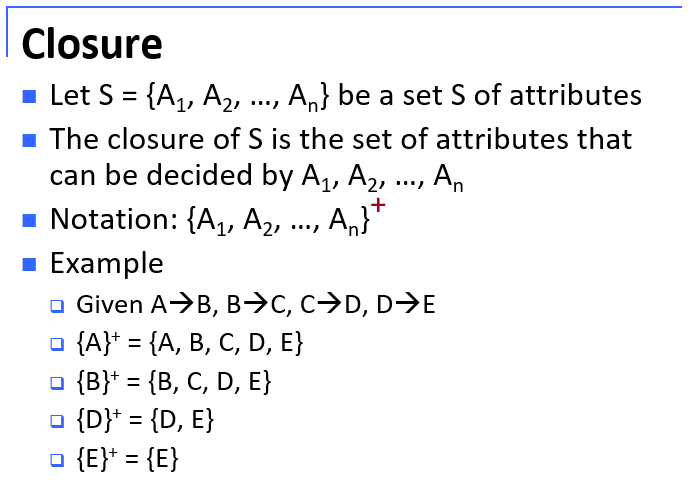
|  |
| --- |
| Purchase( CustomerID, ProductID, ShopID, Price, Date )  Requirement: Each shop can sell at most one product  FD implied: ShopID 🡪 ProductID |
| Purchase( CustomerID, ProductID, ShopID, Price, Date )  Requirement: No two customers buy the same product  FD implied: ProductID 🡪 CustomerID |
| Purchase( CustomerID, ProductID, ShopID, Price, Date )  Requirement: No shop will sell the same product to the same customer on the same date at two different prices  FD implied: CustomerID, ProductID, ShopID, Date 🡪 Price |

|  |  |  |
| --- | --- | --- |
| **Rules** |  | **Example** |
| Axiom of Transitivity | A🡪B, B🡪C then A🡪C | NRIC 🡪 Address, Address 🡪 PostalCode  We have: NRIC 🡪 PostalCode |
| Axiom of Reflexivity | ABCD 🡪 ABC  ABCD 🡪 BCD | NRIC, Name 🡪 NRIC  StudentID, Name, Age 🡪 Name, Age |
| Axiom of Augmentation | Given A 🡪 B, we always have AC 🡪 BC, for any C | Given NRIC 🡪 Name ,  we always have NRIC, Age 🡪 Name, Age |

|  |  |
| --- | --- |
| **Questions** | **Proof** |
| Given A🡪B, BC🡪D  Prove that AC 🡪 D | Given A🡪B, we have AC 🡪 BC (Augmentation)  Given AC🡪BC and BC 🡪 D, we have AC 🡪 D (Transitivity) |
| Given A🡪B, D🡪C  Prove that AD 🡪 BC | Given A🡪B, we have AD 🡪 BD (Augmentation)  Given AD🡪BD, we have AD 🡪 B (Reflexivity)  Given D🡪C, we have AD 🡪 AC (Augmentation)  Given AD 🡪 AC, we have AD 🡪 C (Reflexivity)  In other words, AD decides B and C  Therefore, AD 🡪 BC |
| Given A🡪C, AC🡪D, AD🡪B  Prove that A 🡪 B | Given A 🡪 C, we have A 🡪 AC (Augmentation)  Given A 🡪 AC and AC 🡪 D, we have A 🡪 D (Transitivity)  Given A 🡪 D, we have A 🡪 AD (Augmentation)  Given A 🡪 AD and AD 🡪 B, we have A 🡪 B (Transitivity) |

****

|  |  |
| --- | --- |
| **Questions** | **Proof** |
| Given: A🡪C, C🡪B, B🡪D, D🡪E, E🡪A  Prove C🡪ABE | We start with {C}  Since C🡪B, we have {C, B}  Since B🡪D, we have {C, B, D}  Since D🡪E, we have {C, B, D, E}  Since E🡪A, we have {C, B, D, E, A}  A, B, E are all in the set, so C🡪ABE holds |
| C🡪D, AD🡪E, BC🡪E, E🡪A, D🡪B  Prove C🡪A | We start with {C}  Since C🡪D, we have {C, D}  Since D🡪B, we have {C, D, B}  Since BC🡪E, we have {C, D, B, E}  Since E🡪A, we have {C, D, B, E, A}  A is in the set, so C🡪A holds |
| Given: C🡪D, AD🡪E, BC🡪E, E🡪A, D🡪B, B🡪F  Prove D🡪C | We start with {D}  Since D🡪B, we have {D, B}  Since B🡪F, we have {D, B, F}  What else?  No more.  {D, B, F} is all what can be decided by D  We refer to {D, B, F} as the closure of D |

****

|  |  |
| --- | --- |
| **Questions** | **Proof** |
| A table with six attributes A, B, C, D, E, F  AB🡪C, AD🡪E, B🡪D, AF🡪B | Compute the following closures:  {B, C}+ = {B, C, D}  {A, B}+ = {A, B, C, D, E}  {A, F}+ = {A, F, B, C, D, E} |
| To prove that X 🡪 Y holds, we only need to show that {X}+ contains Y  AB🡪C, AD🡪E, B🡪D, AF🡪B  Prove that AF🡪D  {AF}+ = {AFBCDE}, which contains D  Therefore, AF🡪D holds | |

|  |  |
| --- | --- |
| **Screen Clipping** | **Screen Clipping** |
| A table R(A, B, C)  FDs given: A🡪B, B🡪C   |  |  |  | | --- | --- | --- | | Is A a key?  {A}+ = {ABC}, i.e., A🡪ABC. Yes. | Is B a key?  {B}+ = {BC}, i.e., B does not decide A. No. | Is C a key?  {C}+ = {C}. No. |   Is AB a key?  No, since A is already a key. | |
| **Screen Clipping** | **Screen Clipping** |

To check whether a table is “good”, we need to find the keys of the table

* How do we do that?
* Use functional dependencies (FDs) and closures

|  |  |
| --- | --- |
| Screen Clipping | Screen Clipping |
| **Screen Clipping** | **Screen Clipping** |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Normal Forms:** method to detect “bad tables”, some conditions that a “good” table must satisfy  Various normalforms (in increasing order of strictness)   * First normal form * Second normal form * Third normal form (3NF) * Boyce-Codd normal form (BCNF) * Fourth normal form * Fifth normal form * Sixth normal form  |  |  | | --- | --- | | Screen Clipping | Screen Clipping | | Screen Clipping | Screen Clipping | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **BCNF: Straightforward Checking**  Given: A table R, A set of FDs on R  **Step 1:** Derive the keys of R  **Step 2:** Derive all non-trivial FDs on R   * This is too time-consuming * Trick: Only check the FDs given on R instead of all FDs   **Step 3:** For each non-trivial FD, check if its left hand side contains a key  **Step 4:** If all FDs pass the check, then R is in BCNF; otherwise, R is not in BCNF   |  |  | | --- | --- | | Screen Clipping | Screen Clipping | |  |  | |