



THE UNIVERSITY
of ADELAIDE



CRICOS PROVIDER 00123M

School of Computer Science

COMP SCI 2207/7207 Web and Database Computing

Lecture 28: Schema to SQL

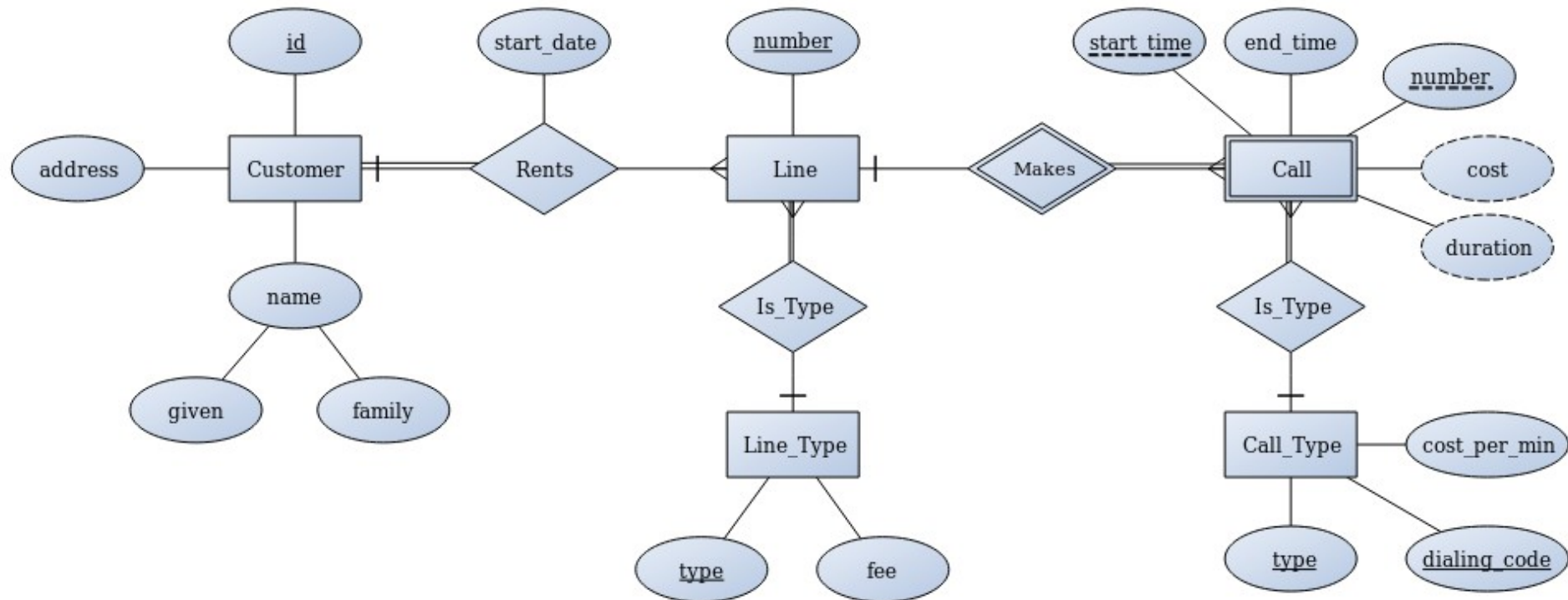
adelaide.edu.au

seek LIGHT

SQL Revisited

- **Create a database**
- **Create tables & constraints**

Phone Company ER Model



SQL Example: Phone Company

- CUSTOMER=(CNumber, Name_first, Name_last, Address)

```
CREATE TABLE Customer
( CNumber INT(11),
  Name_first CHAR(50),
  Name_last CHAR(80),
  Address TEXT
);
```

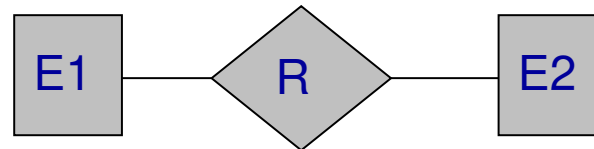
- This is a basic table, but how do add keys and define relationships?

Integrity Constraints

- Purpose:
 - To guard against accidental damage to the database
 - And ensure that changes don't result in a loss of consistency in the data
 - Constrain valid values for data
 - Ensure that related data is deleted from all tables
- Can be any predicate
 - In practice, ones that can be tested efficiently
- Examples
 - Key declaration: definition of primary keys means that updates are constrained
 - Mapping cardinality: constrains the set of relationships

Referential Integrity

- Ensures that values that appear in one table for a given set of attributes also appear for a certain set of attributes in another table
 - Example: If “Hove” is a branch name appearing in one of the tuples in the *account* relation, then there should be a tuple in the *branch* relation for branch “Hove”
- Consider many-to-many relationship between entity sets E_1 and E_2 . The relational schema for R includes the primary keys K_1 of E_1 and K_2 of E_2
 - Then K_1 and K_2 form foreign keys on the relational schemas for E_1 and E_2 respectively



- This provides a constraint between E_1 and R and R and E_2 . If K_1 is removed from E_1 , then it must also be removed from R . Likewise if K_2 is removed from E_2 .

Referential Integrity in SQL

- Primary and foreign keys can be specified as part of the SQL **create table** statement:
 - The **primary key** clause of the **create table** statement includes a list of the attributes that comprise the primary key
 - The **foreign key** clause of the **create table** statement includes both a list of the attributes that comprise the foreign key and the name of the relation referenced by the foreign key

```
CREATE TABLE customer
(customer_number char(10),
customer_name      varchar(20),
customer_address   varchar(100),
PRIMARY KEY (customer_number));
```

```
CREATE TABLE account
(account_number      char(10),
customer_number      char(15),
balance              int(11),
PRIMARY KEY (account_number),
FOREIGN KEY (customer_number) REFERENCES customer(customer_number));
```

Participation Constraints in SQL

- **Partial** participation: Not all entities are involved in the relationship
- Use the **on delete set null** clause to indicate that when the main entity is deleted, the value of the foreign key in the associated entity is set to NULL

```
CREATE TABLE phonenumber (  
    phone_number char(10),  
    customer_number char(10),  
    PRIMARY KEY (phone_number),  
    FOREIGN KEY (customer_number) REFERENCES  
        customer(customer_number) ON DELETE set null );
```


Participation Constraints in SQL

- **Total** participation: All entities in an entity set has to be involved in the relationship set.
 - Example: all ordered items have to be contained in orders
- Use the **on delete no action** clause indicates that the main entity cannot be deleted if it has an association with another entity

```
CREATE TABLE products (  
    product_id INT(11),  
    product_name VARCHAR(50),  
    category VARCHAR(25)  
    PRIMARY KEY (product_id), );  
CREATE TABLE inventory (  
    inventory_id INT,  
    product_id INT NOT NULL,  
    quantity INT,  
    PRIMARY KEY (inventory_id ),  
    FOREIGN KEY (product_id) REFERENCES products (product_id)  
    ON DELETE no action );
```

Weak Entity Constraint in SQL

- A weak entity can be identified uniquely only by considering the primary key of another (strong) entity
- **Primary key:** the primary key of the strong entity, plus partial key
- Use the **on delete cascade** clause indicates that when the strong entity is deleted, all associated weak entities will also be deleted

```
CREATE TABLE orderedItems
    (order_number char(10),
    item_number    char(15),
    item_name      char(10),
    quantity       int,
    PRIMARY KEY (order_number, item_number),
    FOREIGN KEY (order_number) REFERENCES order(order_number)
    ON DELETE cascade);
```

The *check* Clause

- The ***check*** clause permits attributes to be restricted. E.g. an *employee* table:
- Can name constraints
 - The clause “**constraint** *min_wage*” is optional here
 - Useful to indicate which constraint an update violated
 - Otherwise an internally generated id is used
 - Constraints are stored in User_Constraints

```
CREATE TABLE Persons (  
    ID int ,  
    LastName varchar(255) ,  
    FirstName varchar(255),  
    Age int,  
    City varchar(255),  
    CONSTRAINT CHK_Person CHECK (Age>=18 AND City='Adelaide')  
);
```

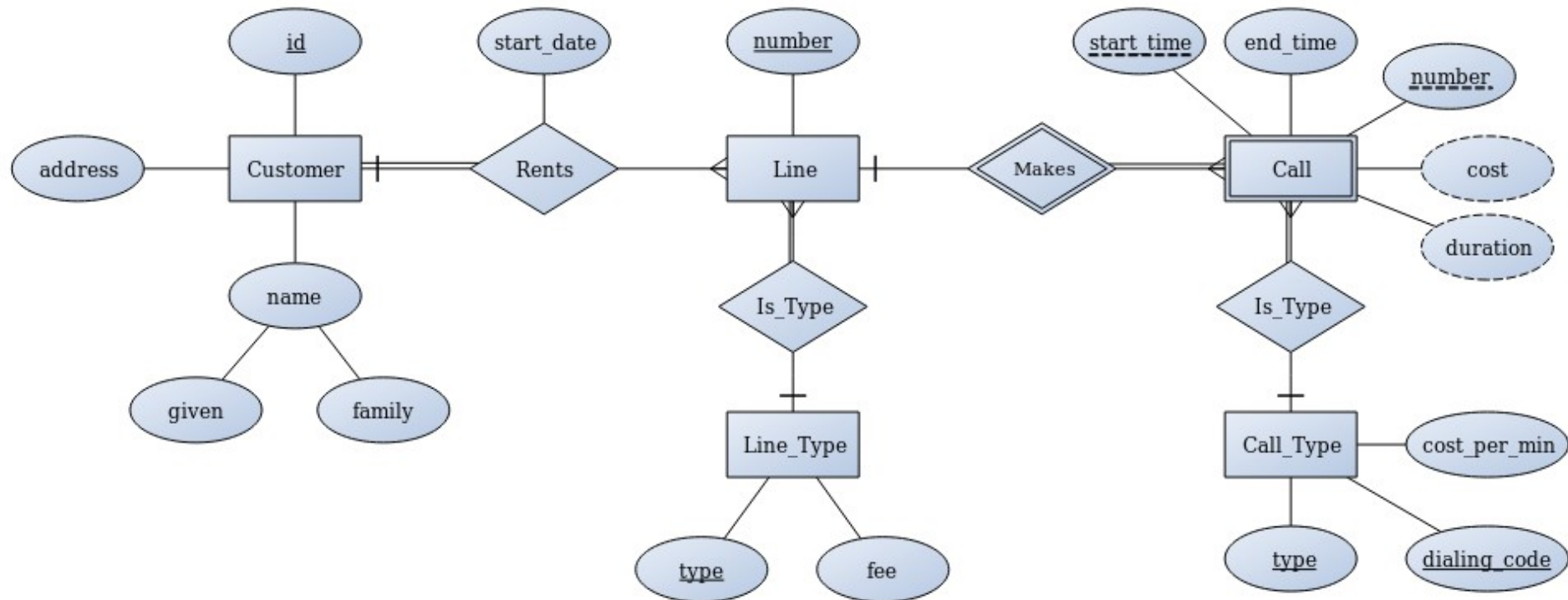
Other constraints

- **UNIQUE** values in column are all different. No duplicates
- **NOT NULL** values in column can not be NULL
- **DEFAULT** sets a default value
- Example:

```
create table account  
  (account_number char(10) UNIQUE,  
  branch_name      char(15) NOT NULL,  
  balance          number(5) DEFAULT(0)  
  );
```

- **PRIMARY KEY** is equivalent to the combination of both **NOT NULL** and **UNIQUE**
- **AUTO_INCREMENT** is used with numeric data to set the value to a number that increases by 1 for each new row

Phone Company ER Model



Phone Company Relation Schema

- Mapping relationships
 - CUSTOMER=(cust_id, name_given, name_family, address)
 - LINE = (number, cust_id, start_date, line_type)
 - CALL_TYPE = (call_type, dialling_code, cost_per_min)
 - LINE_TYPE = (line_type, fee)
 - CALL = (line_number, number, start_time, end_time, call_type)



THE UNIVERSITY
of ADELAIDE



What's Happening?

- Prac Exercise 8 Available, Due Monday Week
 - Websub to come
- Q&A Livestream Tonight 8:30pm
- Keep working on group projects