Lecture 3:
• Relational Model (cont)

Announcements:
• Problem Set 1 (due Thur, Sept 15)

Foreign Keys

<table>
<thead>
<tr>
<th>VehicleType</th>
<th>Vechicle</th>
</tr>
</thead>
<tbody>
<tr>
<td>vt_id</td>
<td>make</td>
</tr>
<tr>
<td>1</td>
<td>Segway</td>
</tr>
<tr>
<td>2</td>
<td>Lime-S</td>
</tr>
<tr>
<td>3</td>
<td>Lime-S</td>
</tr>
</tbody>
</table>

Q: Should the last Vechicle row be legal? ... no vehicle type 4 in DB yet
• represents a “bad” reference

We can prevent these cases using “foreign keys”
• FKs state that a column(s) values must come from a key in another table
• E.g., Vechicle.vt_id should be a foreign key to VechicleType.vt_id
• Requiring Vehicle rows to refer to VechicleType rows (or to be NULL)

Enforcing foreign key constraints helps maintain “referential integrity”
Foreign Keys (cont)

Q: What are the foreign keys in the following tables?

VehicleType(vt_id, make, model, default_plan)
Vehicle(v_id, vt_id, lat, lon)
PricingPlan(p_id, price_per_min, unlock_price)
AllowedPlan(vt_id, p_id)

A: The FKs are:

• Vehicle.vt_id references VehicleType.vt_id
• AllowedPlan.vt_id references VehicleType.vt_id
• AllowedPlan.p_id references PricingPlan.p_id
• VehicleType.default_plan references PricingPlan.p_id

Notes on Foreign Keys

1. A foreign key must reference the entire primary key
   • if R(x, y, z), S(u, x, y), and S has a FK to R
   • then S.(x, y) references R.(x, y) could be a valid FK
   • but S.x references R.x and S.y references R.y are not valid FKs
   • Note: the FK attributes must also have compatible data types (domains)

2. A FK value must contain either NULL or a value of the referenced PK
   • if a Vehicle Type does not have a default pricing plan
   • the row (3, Lime-S, Gen4, NULL) does not violate the FK constraint
   • however, if no PricingPlan row has p.id = 10
   • then (3, Lime-S, Gen4, 10) violates the FK

3. The DBMS enforces FK constraints
   • Thus, DBMSs reject row insertions/updates that violate FKs
Primary and Foreign Keys in Relational Design

Primary and foreign key constraints provide many modeling options ...

Q: For example, what do the following key constraints imply?

1. \text{AllowedPlan}(vt\_id, p\_id)
   - a Vehicle Type can have multiple allowed Pricing Plans
   - a Pricing Plan is allowed for different Vehicle Types

2. \text{AllowedPlan}(vt\_id, p\_id)
   - a Vehicle Type has at most one allowed Pricing Plan
   - a Pricing Plan is allowed for different Vehicle Types

3. \text{AllowedPlan}(vt\_id, p\_id)
   - a Vehicle Type can have many different allowed Pricing Plans
   - any particular Pricing Plan is the allowed plan for at most one Vehicle Type

\textit{Note}: Domains, PKs, and FKs are examples of \textit{“integrity constraints”}
- Other types of integrity constraints are often supported (more later)

Surrogate Keys

\textit{“Naturally” occurring primary keys}
- many tables have attributes that make for "natural" keys (e.g., student id)
- sometimes, the only candidate key is a composite key
- but composite keys aren’t always ideal as foreign keys (e.g., storage space)

\textit{“Artificial” keys are not “naturally occurring” keys}
- added in place of composite keys or if no candidate keys exist
- a \textit{surrogate key} is an artificial key with DBMS-generated values

\textit{When to use surrogate keys ...}
- no natural primary key but need to reference rows
- to "shorten" composite keys
- for latter, can cause uniqueness/consistency issues (more later)
Schema Diagram Basics

1. Each table drawn as a labeled box (with keys designated)
2. Foreign keys as lines between boxes (sometimes linking attributes)

Sometimes called a “table diagram” (but not an “ER diagram”)