

## Moving data between tables

In the last handout we had a table in the Editor's Database:

**Name**(id, name)

in which some ids have multiple names. This complicated certain queries. Suppose we want to create a new table **UName**(id,name) with exactly one (arbitrary) name per id. We do this in two steps:

1. The following SQL command creates the UName table as a copy of the table structure of Name. NO data is added to UName.

```
CREATE TABLE UName like Name;
```

2. The following SQL command combines the INSERT and SELECT commands to put data into UName.

```
INSERT INTO UName (id,name)
SELECT id, min(name)
FROM Name
GROUP BY id
```

The output of the SELECT command is fed into the INSERT command. The field list in the INSERT must match, in number and type of fields, the output from the SELECT.

More complicated SELECT queries can be used to combine data from multiple tables and put it into another table.

## Outer Joins

### Answers

1. Set difference (Name – UniqueName). Use left join to find names not put into UniqueName. First, partial result to show u.name is sometimes null when a record in Name doesn't have a match.

```
select n.id, n.name, u.name  
from aName n left join aUniqueName u using(id,name)
```

Why not use "natural left join"? Because then u.name field is always null (because eliminated) so can't be used for testing for NULL.

Now final result without u.name

```
select n.id,n.name  
from aName n left join aUniqueName u using(id,name)  
where u.name is NULL
```