Inf1B Data and Analysis
Tutorial 2 (week 4)

Hutchins-Korte, Simpson

Question 1 by Byrne (2007)

22 January 2008

- Please answer all questions on this worksheet in advance of the tutorial, and bring with you all work. Tutorials cannot function properly unless you do the work in advance.
- Data & Analysis tutorial exercises are not assessed, but are a compulsory and important part of the course. If you do not do the exercises then you are unlikely to pass the exam.
- Attendance at tutorials is obligatory; please let your tutor know if you cannot attend.
- *Recommended Reading:* Chapter 3 (The Relational Model) and 4 (Relational Algebra) of 'Database Management Systems' (Raghu Ramakrishnan and Johannes Gehrke, 2003). The relevant pages of these chapters can be found on Google Books (`http://books.google.com/`). Furthermore, full copies of Chapter 4 will be handed out during the Thursday lecture (Jan 24th).

1 Introduction

In the previous tutorial, you designed an ER model for a database, based on a description of an inter-university gliding competition scenario. In the first past of this tutorial, you are asked to map the ER model to a relational schema, using the techniques described in the lectures. In the second part of this tutorial, you are asked to formulate some queries on the resulting model using relational algebra. You may wish to refer back to the first tutorial sheet and the information about the gliding competition to remind yourself of particular details such as: which kinds of pilots are allowed to fly solo.

2 Mapping ER to relational schemata

For this question, please use the ER model provided at the end of this tutorial sheet.

Question 1 - Describing Relationships

How is each entity and relationship mapped? State the SQL `create table` statements to define *all* the relations required to create this database. Note that in these SQL statements you will also need to define any *key* and *foreign key* constraints.

3 Queries in Relational Algebra

Consider the following tables describing a fictional inter-university gliding competition (Competition A):

Person

personId	name	experienceLevel
1	John Jones	pre-solo
2	Fraser McEwan	cross-country
3	Jane Smith	instructor
4	John Jones	cross-country
5	Ann Brown	cross-country

Flies

personId	callsign	compDayNo	crewCapacity
1	MF	1	P2
1	MF	2	P1
3	MF	1	P1
2	P19	1	P1
2	P19	2	P1
4	FNS	1	P1
4	FNS	2	P1
5	CPG	1	P1
5	CPG	2	P1

Glider

callsign	type	numSeats
MF	K8	2
P19	Pirat	1
FNS	DG300	2
CPG	Discus	1

Question 2 - Competition A

(a) Using *selection*, specify a query in relational algebra which retrieves all fields in the Person table the for people of at least cross-country standard.

(b) Using *selection* and *projection*, specify a query in relational algebra which retrieves the personId for people of at least cross-country standard.

(c) Using *selection*, *projection*, and *renaming*, specify a query in relational algebra which retrieves the personId for people of at least cross-country standard, renaming 'personId' to 'pilotId'.

Question 3 - Competition B

From now on, we will leave the selection of the appropriate operators up to you. Consider the following tables describing a second fictional inter-university gliding competition (Competition B):

Person

pilotId	name	experienceLevel
1	John Jones	pre-solo
2	Fraser McEwan	cross-country
3	Jane Smith	instructor
6	Fiona Stewart	pre-solo
7	Tom Woods	instructor

Flies

pilotId	callsign	taskNo	crewCapacity
1	MF	1	P2
1	MF	2	P1
3	MF	1	P1
2	P19	1	P1
2	P19	2	P1
6	FNS	1	P2
6	FNS	2	P2
7	FNS	1	P1
7	FNS	2	P1

Glider

callsign	**type**	**numSeats**
MF	K8	2
P19	Pirat	1
FNS	DG300	2

You may assume that both competitions used the same values for personId/pilotId. Furthermore, you may distinguish between tables with the same name using the database prefix. For example, *CompA.Flies* refers to the *Flies* table of Competition A, whereas *CompB.Flies* refers to the `Flies` table of Competition B.

(a) Specify a query in relational algebra which retrieves all fields for all entries in the Flies tables of both Competition A and Competition B.

(b) Specify a query which retrieves the call signs and types of all gliders who were registered for Competition A, but not for Competition B.

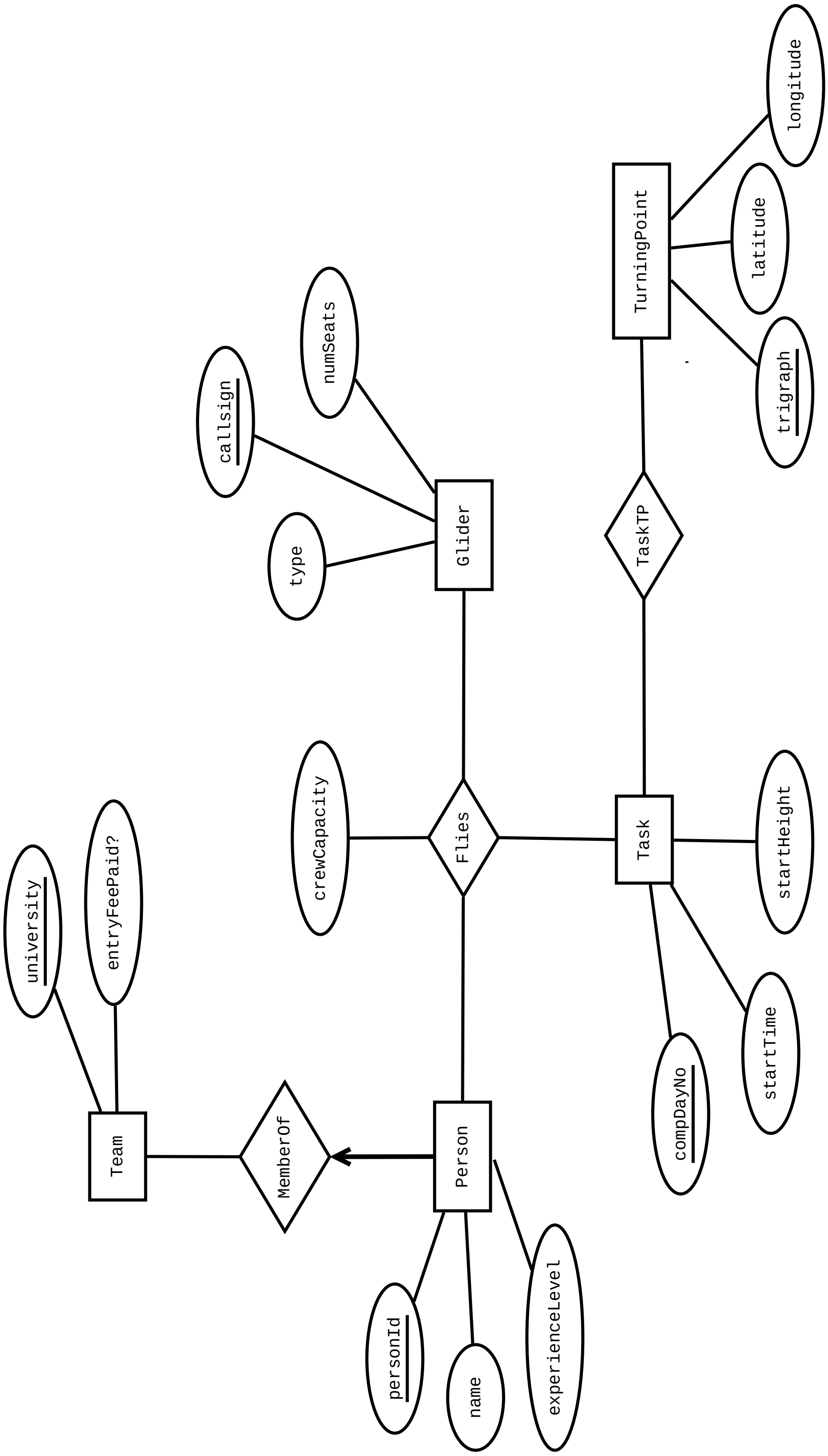
(c) Specify a query which retrieves the call signs and types of all gliders who were registered for both Competition A and Competition B.

(d) To prevent unfair advantages, the organizers of Competition B want to prevent setting a task that some of the participants have already flown in Competition A. Specify a query in relational algebra which retrieves the compDayNos of all the tasks which have been flown by students participating in Competition B.

(e) A gliding inspector wants to check that nobody of pre-solo standard flew solo (or as P1 in a 2-seater) in Competition A. Specify two different queries which both retrieve the personId for all participants of pre-solo standard who flew solo in Competition A. One of your queries should use a *join*, the other should use a *cross-product*.

(f) **(Difficult)** Specify a query which retrieves all possible pairs of pilots (i.e. their personIds) who registered for Competition A and are allowed to fly a two-seater together. Return this as a table with two columns `p1` and `p2`. A tuple should contain as its `p1` value the `personId` of the pilot, and as its `p2` value the `personId` of the copilot.

4 ER Model


longitude
TurningPoint
latitude
trigraph
TaskTP
numSeats
callsign
Glider
type
crewCapacity
Flies
Task
startHeight
university
entryFeePaid?
startTime
compDayNo
Team
MemberOf
Person
personId
name
experienceLevel