

# COMPUTER SYSTEMS – DATABASE ASSIGNMENT

**Name:** Matthew Dyson

**Year:** 1<sup>st</sup> – 2008/2009

**Date:** Sunday, 02 May 2010

**College:** Van Mildert

**User ID:** kklh54

## ABSTRACT

This assignment looks at the input of raw data into an optimised database system in order to achieve the most efficient output. The data provided summarized the organisation of a chain of theme parks in multiple countries, including details of all attractions, employees and ticket sales, albeit in a non-normalized fashion, leading to some data duplication which could eventually cause data loss and inconsistencies.

The first part of the problem was to form a normalized structure for the database, and the output of this is shown in the form of an entities table (showing field descriptions), and then an entity relationship (ER) diagram. After this structure was created, the data provided was interpreted and inserted to the relevant tables, and then multiple queries were executed on the completed database.

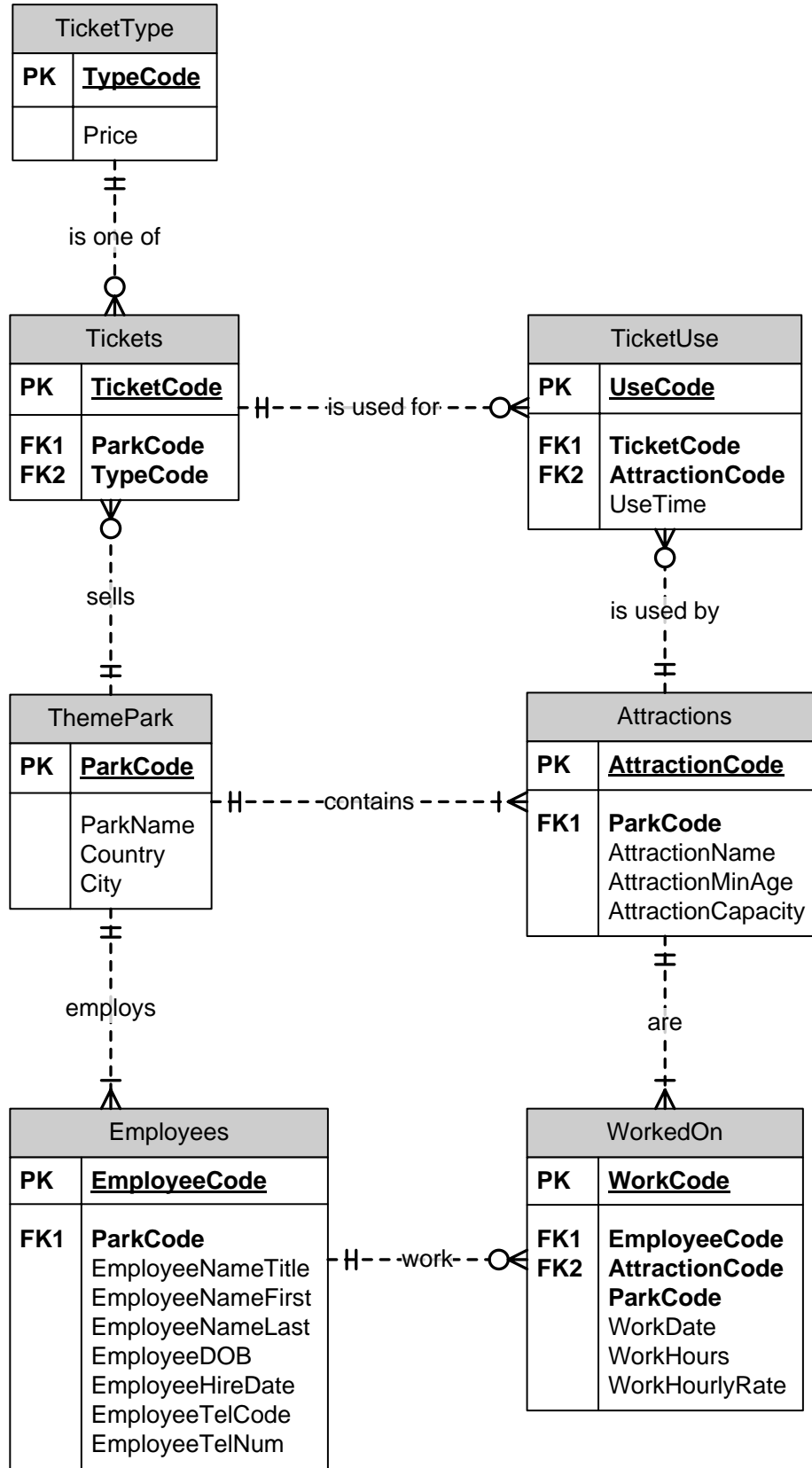
## ER DIAGRAM

The following entities will be necessary in the creation of this database. Primary keys have the field name underlined, and foreign keys have the field name in *italics*.

Table	Field	Description
<b>ThemePark</b>	<u>ParkCode</u>	Unique identifier for the particular park
	ParkName	Name of the park
	Country	Country in which the park is located
	City	City in which the park is located
<b>Attractions</b>	<u>AttractionCode</u>	Unique identifier for the attraction
	AttractionName	Name of the attraction
	<i>ParkCode</i>	Appropriate park code for the attraction
	AttractionMinAge	Minimum age for the attraction
	AttractionCapacity	Capacity of the attraction
<b>Employees</b>	<u>EmployeeCode</u>	Unique code for the employee
	<i>ParkCode</i>	Park code that the employee works in
	EmployeeNameTitle	Employees title (Mr, Mrs etc)
	EmployeeNameFirst	Employees first name
	EmployeeNameLast	Employees last name

Table	Field	Description
	EmployeeDOB	Date of birth of the employee
	EmployeeHireDate	Date that the employee was hired
	EmployeeTelCode	Employees telephone area code
	EmployeeTelNum	Employees telephone number
<b>WorkedOn</b>	<u>WorkCode</u>	Unique identifier for the link table
	EmployeeCode	Employee code for the worker
	AttractionCode	Attraction code
	WorkDate	Date on which the employee worked on the attraction
	WorkHours	Hours which the employee worked
	WorkHourlyRate	Pay per hour for the employee
<b>Tickets</b>	<u>TicketCode</u>	Unique code for the ticket
	TypeCode	The type of ticket
	ParkCode	The park that the ticket applies to
	DateSold	Date that the ticket was sold on
<b>TicketType</b>	<u>TypeCode</u>	Unique identifier for this type of ticket
	Price	Price of the ticket
<b>TicketUse</b>	UseCode	Unique identifier for the link table
	TicketCode	The ticket for the use
	AttractionCode	The attraction that is being used
	UseTime	The time of the use

The following diagram shows how these entities will be related.



## CONCLUSIONS

Within this assignment, I have researched and used areas of SQL that I had not touched on before, in order to simplify the SQL statements above. The alternative to the solutions I have found would often cause a massive load on the server, although my answers may well be taken further!

One such example of this is the use of 'queries within queries' in question 6 where the use of a nested SELECT COUNT statement allows for the efficient calculation of the average. One alternative to this is the use of variables within the statements, which could have been utilized in this question. I did use variables in solving question 11, where it is necessary to find Arif Arshad's employee code (his unique identifier) before querying this against another table. This method could also be applied to some other questions in order to find primary keys from other identifying information, although one must be careful to ensure that a single row is selected for input into the variable, hence use of the LIMIT clause.

One other item of note is the use of AS within some of the SELECT statements. This is mainly for aesthetic reasons, and is not dissimilar to use of meaningful variable names within programming – it just makes it easier to read the output from the SQL parser.

## REFERENCES

- <http://www.w3schools.com/SQL>  
Function syntax
- <http://lists.mysql.com>  
Function syntax