SQL-EDDI
A computer-based environment for the
study of relational query languages

Meurig Beynon, Abhir Bhalerao, Chris Roe, Ashley Ward
Department of Computer Science
University of Warwick, Coventry, UK

http://www.dcs.warwick.ac.uk/modelling
http://empublic.dcs.warwick.ac.uk/projects/sqleddiWard2003

Contents of talk

Educational objectives of SQLEDDI

Practical details of SQLEDDI

Development of SQLEDDI

Conclusions and future work

Educational objectives of SQLEDDI

Codd's relational theory: algebraic and logical foundations for RDBs

Foundation for relational query languages ...

Information Systems Base Language (ISBL)
Stephen Todd, IBM Peterlee, 1976

Closely based on relational algebra ...

About ISBL

Primitive operations of relational algebra

union (+)
intersection (.)
difference (-)
natural join (*)
selection (:)
projection (%)

Interrogate expression   execute a query

Assign value to variable   define a table

Assign formula to variable   define a view

EDDI implements ISBL as a 'definitive notation'

```
%eddi

allfruits (name CHAR, begin INT, end INT);
allfruits << ["granny",8,10],["lemon",5,12],["kiwi",5,6],["passion",5,7];
allfruits << ["orange",4,11],["grape",3,6],["lime",4,7],["pear",4,8];

apple (name CHAR, price REAL, qnt INT);
apple << ["cox",0.20,8],["red",0.35,4],["granny",0.25,20];

citrus (name CHAR, price REAL, qnt INT);
citrus << ["lime",0.30,3],["orange",0.65,8],["kiwi",0.75,5],["lemon",0.50,2];

soldfruit (name CHAR, unitsold INT);
soldfruit << ["cox",100],["granny",153],["red",70];
soldfruit << ["kiwi",23],["lime",15],["lemon",55],["orange",78];

fruits is allfruits % name;
popcitrus is (fruits.citrus % name).(soldfruit : unitsold > 50 % name);
nonapplesncox = allfruits-(allfruits*apple%name,begin,end)+allfruits:name=="cox";
```

EDDI - base for teaching environment used at Warwick over 4 years:
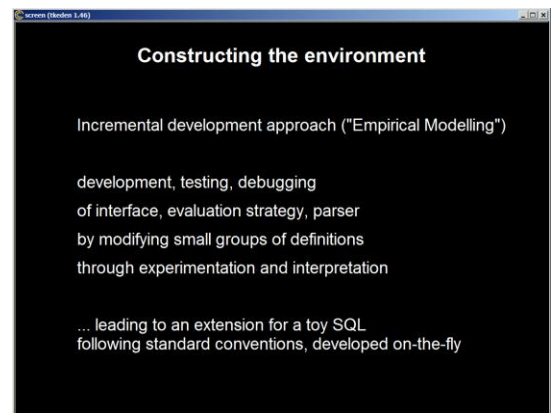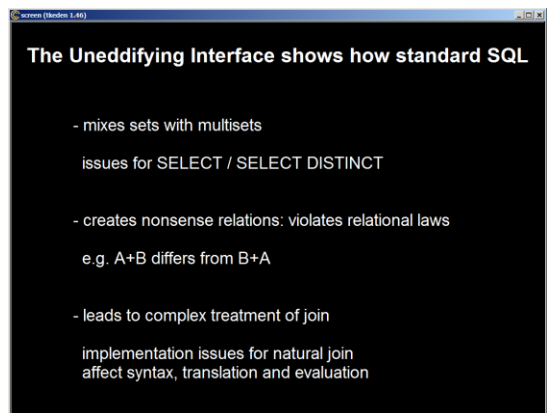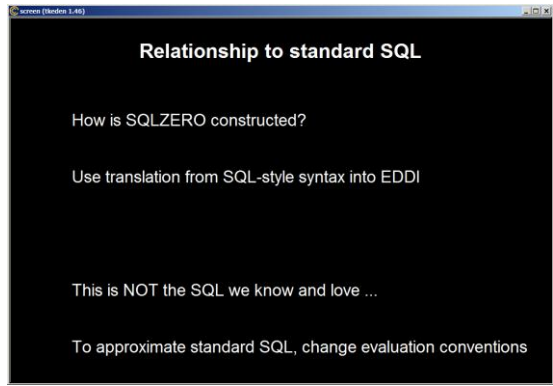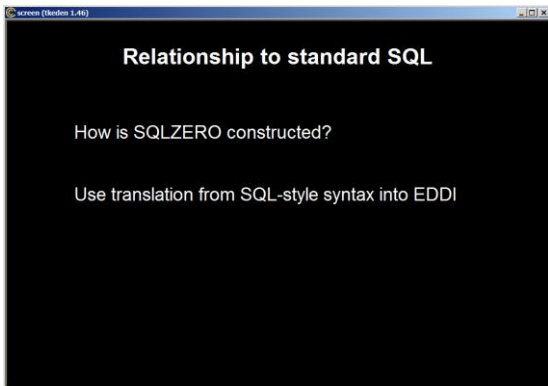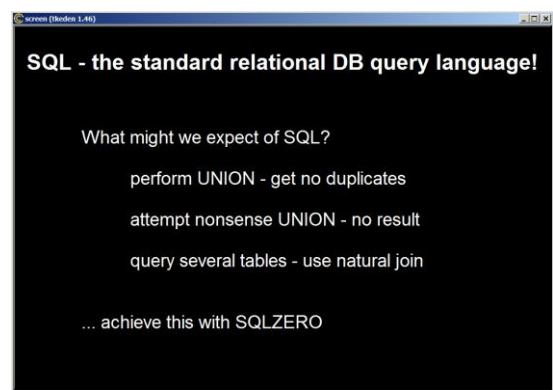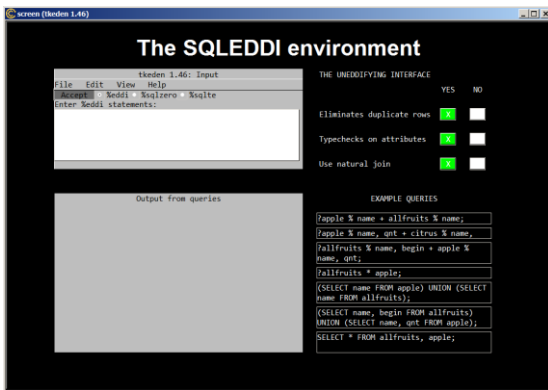approximately 300 computing-based 2nd year students each year

Illustrate a simplified version in this talk ...

Mathematical foundation for EDDI

set-theoretic operations: union of relations

strict type checking: domains + attributes

pure relational algebra operators: natural join

The SQLEDDI environment

SQL - the standard relational DB query language!

What might we expect of SQL?

perform UNION - get no duplicates

attempt nonsense UNION - no result

query several tables - use natural join

... achieve this with SQLZERO

Relationship to standard SQL

How is SQLZERO constructed?

Use translation from SQL-style syntax into EDDI

Relationship to standard SQL

How is SQLZERO constructed?

Use translation from SQL-style syntax into EDDI

This is NOT the SQL we know and love ...

To approximate standard SQL, change evaluation conventions

The Uneddifying Interface shows how standard SQL

- mixes sets with multisets

   issues for SELECT / SELECT DISTINCT

- creates nonsense relations: violates relational laws

   e.g. A+B differs from B+A

- leads to complex treatment of join

   implementation issues for natural join
   affect syntax, translation and evaluation

Constructing the environment

Incremental development approach ("Empirical Modelling")

development, testing, debugging
of interface, evaluation strategy, parser
by modifying small groups of definitions
through experimentation and interpretation

... leading to an extension for a toy SQL
following standard conventions, developed on-the-fly

Further potential applications

- studying query optimisation

- other query languages (e.g. QUEL)

- other aspects of SQL (e.g. DDL)

- assessment environment

- integrated teaching vehicle

- visualisation to assist learning and evaluation

The Temposcope

Conclusions

- interest and understanding of flaws of SQL

- generalisation of relational modelling

http://www.dcs.warwick.ac.uk/modelling

http://empublic.dcs.warwick.ac.uk/projects/sqleddiWard2003