

SQL in de klas

Een experiment met probleemoplossen in het informatica-onderwijs

B. van Dijk

Faculteit Informatica, UT Twente

Samenvatting

In dit artikel wordt gerapporteerd over experimentele lessen 'probleemoplossen met SQL', en de ervaringen die daarmee zijn opgedaan in een tweetal proefscholen. Voor de cursus is lesmateriaal ontwikkeld; een voorbeeld wordt gepresenteerd.

Inleiding

De cursus waarvan in dit artikel verslag wordt gedaan is ontwikkeld als een poging om probleemoplossen tot een centraal onderdeel van informaticalessen te maken. Die poging is niet nieuw: in het verleden is op grote schaal geëxperimenteerd met probleemoplossen met behulp van programmeren in een imperatieve taal (Basic, Pascal). De resultaten van die experimenten zijn teleurstellend, vooral omdat voor programmeren in een imperatieve taal vrij veel kennis nodig is van taalelementen en van de werking van computers. Kennis die kennelijk niet in korte tijd beheerst kan worden.

In het voortgezet onderwijs is een beperkte hoeveelheid tijd beschikbaar voor het vak informatica. In de eerste twee jaar zal het vak informatiekunde worden gegeven; hiermee is al ruimschoots geëxperimenteerd, en de eindtermen zijn in ontwikkeling. Aan dit vak mogen in totaal twintig lesuren worden besteed. Voor de middenbouw (3e en/of 4e klas van havo/vwo) is één uur per week gedurende één jaar beschikbaar, wat onvoldoende tijd is om te leren programmeren in een imperatieve taal.

Om in betrekkelijk weinig tijd serieus aandacht aan probleemoplossen te kunnen schenken, moet iets gevonden worden dat imperatieve talen kan vervangen. Een bijkomende overweging daarvoor is dat het gebruik van imperatieve talen terugloopt, terwijl de vaardigheid om grote hoeveelheden gegevens te kunnen hanteren steeds belangrijker wordt. In dit licht is de toegenomen belangstelling voor databases begrijpelijk.

In het hier beschreven experiment is gekozen voor probleemoplossen met behulp van de databasevraagtaal SQL.

Probleemoplossen en SQL

Databases lenen zich goed voor het ontwikkelen van probleemoplosvaardigheden, zeker wanneer een vraagtaal gebruikt wordt die 'eenvoudig' te leren valt. De taal SQL is zo ontworpen dat ook gebruikers met weinig informaticakennis ermee kunnen leren werken. Toepassing ervan vindt in toenemende mate plaats, met name ook buiten het onderwijs.

SQL is een zogenaamde relationele databasevraagtaal, dat wil zeggen dat gegevens steeds worden gepresenteerd in de vorm van tabellen. Als voorbeeld geef ik een paar tabellen (in verkorte vorm) die afkomstig zijn uit een van de databases in het lesmateriaal, een database waarin gegevens van een schooladministratie zijn opgenomen:

Tabel LEERLING

LLNR	NAAM	KLAS
1345	PIETERS RENE	ATH 4B
1099	ERNSTINE	HAVO 3D
1052	MASTMAAIKE	ATH 4B
1801	ARENDS ANS	HAVO 3D
1952	BAVINKJAN	HAVO 3D
1419	LOPEZJUAN	GYM 2A

Tabel LERAAR

LCODE	LNAAM	LADRES	LWOONPL	KLAS[I]
PT	PAAPSTR	BORCHERT 8	ENSCHEDÉ	GYM 2A
KU	KUSTERS I	MAARDIJK 9	BORNE	-
JR	JAGER W	DORPEL 9	NEEDE	ATH 4B
YZ	YUMAZ T	ESCHERSTR 6	ENSCHEDÉ	HAVO 3D

In de SQL-cursus komen verschillende aspecten van probleemoplossen aan de orde. In de eerste plaats wordt een in gewoon Nederlands gestelde vraag zo bondig mogelijk herformuleerd, waarbij duidelijk dient te worden welke gegevens nodig zijn en welke voorwaarden daaraan nog gesteld moeten worden. Zo'n analyse en specificatie van een gestelde vraag vormt de belangrijkste fase in het probleemoplossingsproces.

Een volgend aspect van probleemoplossen betreft het werken met gegevens die verspreid zijn over verschillende tabellen. Bij het opvragen van gegevens moet dan bepaald worden uit welke tabel of tabellen geselecteerd moet worden en hoe koppeling van die gegevens tot stand kan worden gebracht.

Relationele databases bestaan gewoonlijk uit verschillende tabellen. Door met meer tabellen te werken kan grotendeels vermeden worden dat gegevens meer dan één keer moeten worden ingevoerd. Ter illustratie volgt de tabel die ontstaat als de twee bovenstaande tabellen tot één tabel zouden worden samengevoegd.

Samengevoegde tabel

LIJNR	NAAM	KLAS	LCODE	LNAAM	LADRES	LWOONPL
1345	PIETERSRENE	A1114B	JR	JAGERW	DORPEL9	NEEDE
1099	ERNSTINE	HAVO3D	YZ	YUMAZT	ESCHERSTR6	ENSCHIEDE
1052	MASTMAAIKE	A1114B	JR	JAGERW	DORPEL9	NEEDE
1801	ARENDS ANS	HAVO3D	YZ	YUMAZT	ESCHERSTR6	ENSCHIEDE
1952	BAVINR JAN	HAVO3D	YZ	YUMAZT	ESCHERSTR6	ENSCHIEDE
1419	LOPEZJUAN	GYM2A	PT	PAAPSTR	BORCHERTS	ENSCHIEDE

Bij elke leerling zijn nu de gegevens van de klasleeraar vermeld. Dit heeft als gevolg dat gegevens van sommige leraren (de klasleeraren) vaak herhaald worden, terwijl de gegevens van overige leraren niet voorkomen.

Een derde aspect van probleemoplossen betreft het bewerken van gegevens. Een vraag die zo'n bewerking nodig maakt is bijvoorbeeld: 'Verzuimen jongens meer dan meisjes?' Voor een antwoord op deze vraag kan niet volstaan worden met onbewerkte gegevens uit de database zelf; in dit geval moet het gemiddelde verzuim van de jongens vergeleken worden met dat van de meisjes.

Het lesmateriaal

In het lesmateriaal, ontwikkeld voor een serie van ongeveer vijftien lessen, wordt vrijwel alleen aandacht besteed aan het opvragen van informatie uit een bestaande database.

De volgende onderwerpen worden behandeld:

- De begrippen database/gegevensbank, tabel, kolom, rij, kenmerk en waarde van een kenmerk.
- Het opvragen van gegevens uit één tabel. Deelonderwerpen: sorteren, het selecteren van kolommen en rijen uit een tabel, het werken met logische operatoren (AND, OR, NOT) in samengestelde voorwaarden.
- Ingebouwde functies (COUNT, MAX, MIN, SUM, AVG).
- Groeperen (GROUP BY).
- Het werken met verschillende tabellen, door mid-

del van subvragen en door middel van het samenvoegen van tabellen ('join').

- Het toevoegen, wijzigen en verwijderen van rijen uit een tabel.

Hiermee zijn de taalelementen van SQL niet allemaal aan bod gekomen. Het leren probleemoplossen staat immers centraal, niet het leren van zoveel mogelijk taalelementen van SQL.

In grote lijnen is het lesmateriaal als volgt opgebouwd. Elk nieuw onderwerp wordt uitgelegd aan de hand van een voorbeeld, waarin een oplossing voor een probleem wordt afgeleid. De eindoplossing zal in het algemeen een SQL-commando zijn.

De leerlingen krijgen na het lezen van het voorbeeld de opdracht om het commando door de computer te laten uitvoeren, waarna ze de uitkomst moeten controleren. Tot slot worden nog een paar vergelijkbare problemen aangeboden die door de leerlingen moeten worden opgelost. In het algemeen staat op het eind van een hoofdstuk een samenvatting, gevolgd door een paar opdrachten.

Bij het theorie-dictaat horen werkbladen die tijdens het maken van de opdrachten moeten worden ingevuld en een inlegvel waarop de belangrijkste commando's, de werking van functie- en andere speciale toetsen, en de inhoud van de in het lesmateriaal gebruikte voorbeelddatabases zijn aangegeven.

De opbouw van het lesmateriaal illustreren we met een (enigszins verkort) gedeelte [2] waarin het werken met samengestelde voorwaarden wordt behandeld:

Bea leert op een feestje een leerling van de Escherschool kennen.

Later bedenkt ze dat ze hem nog wel eens wil ontmoeten. Helaas weet ze zijn adres niet. Wel kan ze zich herinneren dat hij Vermeer heet en dat hij niet in Enschede of Hengelo woont. We gaan proberen het adres van deze jongen op te sporen.

We willen dus weten: adres, postcode en woonplaats van leerling Vermeer. Om het antwoord te kunnen controleren én omdat Bea toch ook wel graag de voornaam van de jongen zal willen weten, vragen we ook de naam op.

Al deze gegevens staan in de tabel LEERLING.

De voorwaarde luidt:

V: leerling is een jongen die Vermeer heet en die niet in Enschede of Hengelo woont

We zetten dit in een schema:

KENMERKEN: naam, adres, postcode, woonplaats
TABEL: LEERLING
VOORWAARDE: V

MET V: leerling is een jongen die Vermeer heet en die niet in Enschede of Hengelo woont

In V herkennen we drie afzonderlijke voorwaarden:

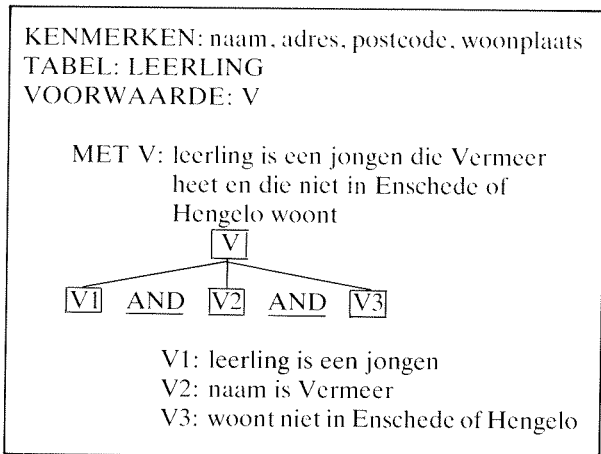
- V1: leerling is een jongen
- V2: naam is Vermeer
- V3: woont niet in Enschede of Hengelo

Deze voorwaarden moeten alle drie gelden. De samengestelde voorwaarde V wordt dus:

V1 AND V2 AND V3

Aangezien de voorwaarde V1, V2 en V3 alle drie duidelijk afzonderlijke voorwaarden zijn (we kunnen niet twee van deze voorwaarden als één geheel beschouwen) hebben we geen haken gezet.

We breiden het schema weer uit:



We werken V1, V2 en V3 nog uit.

Voorwaarden V1 en V2 zijn enkelvoudig en hoeven dus hier niet meer verder te worden uitgewerkt. Voorwaarde V3 is ingewikkelder. We kunnen deze op twee manieren verder uitwerken:

1. De voorwaarde heeft de vorm NOT V4, waarbij V4 de voorwaarde is waarmee de leerlingen geselecteerd worden die wél in Enschede of Hengelo wonen.

Of:

2. We kunnen de voorwaarde herformuleren tot: 'woont niet in Enschede en woont niet in Hengelo'.

We werken hier manier 1 uit. De voorwaarde heeft in dit geval de vorm:

NOT V4

We zoeken immers leerlingen die niet voldoen aan de voorwaarde:

V4: woont in Enschede of Hengelo

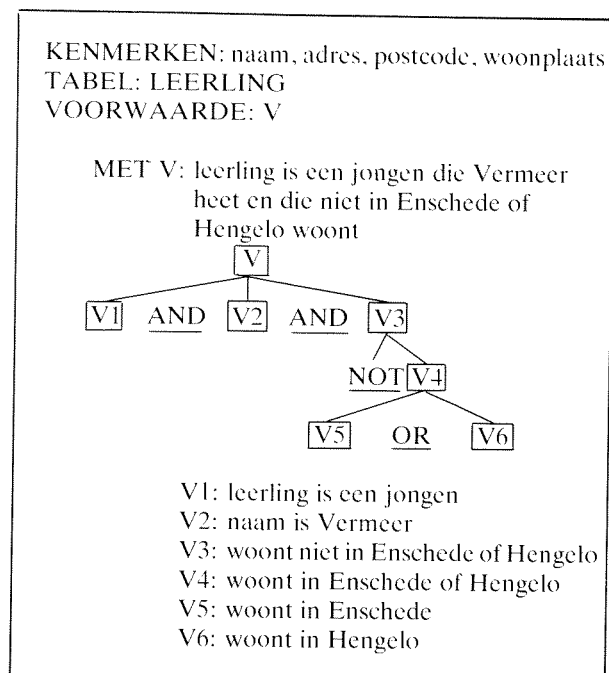
Vervolgens werken we V4 verder uit. We herkennen in V4 de voorwaarden:

- V5: woont in Enschede
- V6: woont in Hengelo

Welke logische operator is nu van toepassing? Een leerling voldoet aan voorwaarde V4, als deze ofwel in Enschede woont ofwel in Hengelo. De operator is dus OR, en V4 is gelijk aan:

V5 OR V6

We kunnen nu het schema aanvullen:



Voorwaarden V5 en V6 zijn enkelvoudig. We stoppen dus met het uitbreiden van het schema.

We vertalen de gegevens uit het schema weer in een SQL-commando:

```

SELECT NAAM, ADRES, PSTCODE, WOONPL -
FROM LEERLING -
WHERE GESL='M' -
AND -
NAAM LIKE 'VERMEER%' -
AND -
NOT (WOONPL='ENSCHEDA' OR WOONPL='HENGELO')

```

Let op de haakjes in de voorwaarde. Hiermee geven we weer aan dat de samengestelde voorwaarde WOONPL='ENSCHEDA' OR WOONPL='HENGELO' als één voorwaarde (V4) moet worden opgevat en dat NOT betrekking heeft op deze hele samengestelde voorwaarde. Als de haakjes worden weggelaten heeft NOT alleen betrekking op de voorwaarde die er onmiddellijk achter staat (in dit geval: WOONPL='ENSCHEDA').

Opdracht 1

- a. Voer het commando uit en noteer de geselecteerde naam (of namen) en de woonplaats(en) op het werkblad.
- b. Haal het commando terug met SHIFT F2 en verwijder de haken in de laatste regel van het commando. Voer dit nieuwe commando uit en vul het werkblad in.

Opmerking:

We hebben in deze paragraaf een tweede manier genoemd om voorwaarde V3 uit te werken. Deze tweede manier werken we hier niet helemaal meer uit. We geven alleen nog de voorwaarde die we op die andere manier zouden vinden:

```

(NOT WOONPL='ENSCHEDA') AND
(NOT WOONPL='HENGELO')

```

Hoewel deze voorwaarde er heel anders uitziet, geeft het invullen van deze voorwaarde in plaats van de eerder gevonden voorwaarde:

```
NOT (WOONPL='ENSCHEDÉ' OR  
WOONPL='HENGÉLO')
```

toch hetzelfde resultaat.

Opdracht 2

De feeste commissie van de Escher School heeft leerlingen nodig om te assisteren bij het jaarlijkse schoolfeest op 15 november. Omdat het misschien laat zal worden, zeker voor de 'opruimers', wil men liefst wat oudere leerlingen vragen.

In eerste instantie wordt daarbij gedacht aan de leerlingen uit de zesde klas, én alle leerlingen die op 15 november 1988 ouder zijn dan 18 jaar. Verder komen ook in aanmerking: leerlingen uit Enschede die op 15 november 1988 ouder zijn dan 17 jaar.

- Schrijf het selectie-commando waarmee een lijst kan worden opgevraagd waarop naam, woonplaats, klas en geboortedatum staat van alle leerlingen die hiervoor staan omschreven. Gebruik daarbij het werkblad.
- Voer het commando uit en controleer het resultaat. Vul het werkblad in.

Het experiment

In het voorjaar van 1988 is een eerste versie van het lesmateriaal uitgetest in twee 4vwo klassen die één lesuur per week hiermee werkten. De leerlingen waren verplicht bij de lessen aanwezig, maar kregen geen cijfer.

In de eerste les werd in het kort het doel van het onderzoek toegelicht. De leerlingen werden vervolgens in groepjes van drie ingedeeld, en kregen de volgende instructies:

- Bestudeer het dictaat zo zelfstandig mogelijk; stel eventuele vragen pas aan de leraar nadat je eerst geprobeerd hebt om zelf, samen met andere leden van je groep, het antwoord te vinden. Los ook de opdrachten zoveel mogelijk zelfstandig (groepsgevijs) op.
- Roep in geen geval de hulp in van leden van een andere groep.
- Wissel regelmatig van plaats, zodat iedereen af en toe achter het toetsenbord van de computer zit.

De docenten werd gevraagd om alleen op vragen te reageren en dan met minimale hulp de leerlingen weer op weg te helpen. Klassikale uitleg werd nooit gegeven. De groepjes werkten het lesmateriaal in eigen tempo door.

De samenwerking in de groepjes was in het algemeen goed. Over de op te lossen problemen werd volop gediscussieerd. Ook kwam het herhaaldelijk voor dat leerlingen elkaars fouten corrigeerden, of de uitkomst van een commando aan elkaar voorspelden, nog voordat deze op het scherm verscheen.

De groepjes werkten zeer zelfstandig, zodat de docenten weinig werk hadden.

Aan het eind van het schooljaar werd een eindtoets afgenomen, waarbij een database bevroegd moest worden die niet in het lesmateriaal behandeld was. Een omschrijving van deze database (in de vorm van voorbeeldtabellen waarin een paar rijen waren ingevuld) werd tegelijk met de toetsvragen uitgereikt. Omdat op dat moment de klassen geen vijftien lessen met het lesmateriaal gewerkt hadden, maar elf respectievelijk dertien lessen, waren veel groepjes niet toegekomen aan de laatste hoofdstukken waarin het werken met verschillende tabellen behandeld wordt. Over het werken met verschillende tabellen wordt hier dan ook niet gerapporteerd.

De toets werd in het algemeen goed gemaakt: aspecten van een oplossing die in vrijwel elke opdracht terugkeren, zoals het opvragen van kenmerken, de tabelkeuze en het correct formuleren van enkelvoudige voorwaarden, bleken door vrijwel alle leerlingen goed beheerst te worden. De meer eenvoudige opgaven in de toets werden dan ook erg goed gemaakt. Moeilijker bleek het kiezen van functies (vooral SUM en COUNT werden vaak verwisseld), en het construeren van samengestelde voorwaarden. Bij samengestelde voorwaarden moeten logische operatoren (AND, OR en NOT) gebruikt worden en vaak ook haken. Fouten bij de keuze van logische operatoren lijken samen te hangen met de formulering van vragen, zoals duidelijk wordt als we toetsopgaven 3 en 5 met elkaar vergelijken:

- Schrijf een SQL-commando om de nummers op te vragen van de vliegtuigen die op 29 of 30 juni 1988 naar Ankara vliegen.

Antwoord:

```
SELECT VLIEGTUIGNR  
FROM VLUCHT  
WHERE (DATUM='880629' OR  
DATUM='880630') AND BESTEMMING=  
'ANKARA'
```

- Schrijf een SQL-commando om het bedrag op te vragen dat betaald moet worden door iemand die met vluchtnummer 714 van Amsterdam naar Brussel vliegt en daarna met vluchtnummer 173 van Brussel naar Malaga.

Antwoord:

```
SELECT SUM(PRIJS)  
FROM VLUCHT  
WHERE VLUCHTNR='714' OR  
VLUCHTNR='173'
```

Terwijl in opgave 3 door 93% van de leerlingen de operator OR goed gekozen is, kiest 63% bij opgave 5 ten onrechte de operator AND.

De haken om samengestelde voorwaarden werden vaak ten onrechte weggelaten. Het kwam nauwelijks voor dat haken wel aanwezig waren, maar op de verkeerde plaats.

Het werken met SQL is inhoudelijk sterk gebaseerd op logica en verzamelingenleer. Hoewel in het lesmateriaal de associatie met wiskunde niet expliciet ge-

maakt is (teneinde leerlingen die zwak zijn in wiskunde geen negatieve starthouding te bezorgen), verwacht ik verband tussen wiskundecijfers en de prestaties op de SQL-toets. Dit verband was inderdaad statistisch significant.

Een dergelijk verband had ik ook verwacht tussen de cijfers voor tekstverklaring en prestaties op de SQL-toets. Om een commando te kunnen schrijven dat precies de gewenste informatie oplevert moeten leerlingen immers een informatievraag goed kunnen lezen en herformuleren. Uit cijfers bleek echter geen verband tussen de SQL-eindtoets en tekstverklaring.

Aan de leerlingen werd gevraagd aan te geven of en hoe ze al eerder met computers in aanraking waren gekomen, dit om na te gaan of de resultaten op de eindtoets verband hielden met hoeveelheid computerervaring. Opmerkelijk was het feit dat er geen aantoonbaar verband bestond.

Door de leerlingen is ook een evaluatieformulier ingevuld met vragen die voornamelijk betrekking hadden op de lessen zelf: of ze die lessen leuk, moeilijk, afwisselend vonden.

Op de vraag of ze het volgen van de lessen leuk vonden, scoorden de leerlingen gemiddeld 'niet leuk, niet vervelend'; het verband met het resultaat op de eindtoets was positief: leuk vinden en een hoog cijfer gaan samen.

Het moeilijk vinden van de leerstof correleert eveneens met de score op de eindtoets: wie de stof moeilijk vindt, scoort laag. (Gemiddeld vond men de moeilijkheidsgraad van de stof 'net goed'.)

Interessant is dat 'moeilijk vinden van opdrachten' veel minder sterk verband houdt met de score op de eindtoets. Ik veronderstel dat een deel van de leerlingen zich een verkeerd beeld heeft gevormd van de moeilijkheid van de opdrachten, in de zin dat aangeboden problemen niet of onvoldoende doorzien werden. Deze veronderstelling verwijst naar de noodzaak van voldoende terugkoppeling op de door de leerlingen gevonden oplossingen.

Verder bleek uit de evaluatie dat de leerlingen de lessen in het algemeen als niet bijzonder afwisselend beschouwden. Deels op basis van antwoorden op enkele open vragen op het evaluatieformulier kunnen drie mogelijke oorzaken genoemd worden:

1. In alle lessen werd maar één werkvorm gebruikt: leerlingen werkten, zittend achter de computer, het lesmateriaal door; dit lesmateriaal heeft een vaste structuur: tekst lezen, voorbeeldcommando uitvoeren, uitvoer controleren, opdrachten maken, invullen op werkblad en uitvoeren, de uitvoer controleren.
2. In de lessen werden alleen databases bevráágd; voor het bevráagen van databases worden in SQL steeds dezelfde sleutelwoorden gebruikt: SELECT, FROM, WHERE. Bij sommige leerlingen lijkt deze eigenschap van SQL, dat weinig verschillende taalelementen gebruikt hoeven te worden, het gevoel op te roepen dat ze weinig leren.

3. In het lesmateriaal wordt voornamelijk gebruik gemaakt van één enkele database, de database van een schooladministratie. De behoefte om te werken met meer verschillende databases wordt waarschijnlijk nog versterkt door het feit dat het merendeel van de leerlingen deze database niet interessant vond.

In de evaluatie werd ook gevraagd aan welke groeps-grootte de leerlingen de voorkeur zouden geven bij het werken met SQL. Ruim een kwart van de leerlingen gaf te kennen liefst alleen te werken, terwijl ongeveer de helft de voorkeur gaf aan groepjes van twee. Verder wilde 16% werken in groepjes van meer dan twee personen; de rest had geen mening.

Besluit

De resultaten van de leerlingen, na afloop van de (niet eens volledig afgemaakte) cursus, waren bevredigend. Vrijwel alle leerlingen waren in staat om elementaire SQL-commando's foutloos of bijna foutloos te schrijven.

Niettemin valt de cursus te verbeteren. Door meer verschillende werkvormen te gebruiken, met name ook af en toe klassikale uitleg, komt er meer afwisseling in de lessen. Deze afwisseling kan ook groter gemaakt worden door meer verschillende databases te gebruiken, en door behalve het bevráagen van een database ook het ontwerpen ervan in de cursus op te nemen. Een cursus van vijftien lessen lijkt daarvoor dan wel wat kort.

Tijdens het experiment hebben de docenten zich steeds zeer terughoudend opgesteld. Dit was aan hen gevraagd, om te voorkomen dat gebreken in het lesmateriaal door docenten zouden worden opgevangen; het materiaal zelf moest immers getest worden. In een reguliere cursus heeft een grotere inbreng van docenten naar verwachting positieve effecten: niet alleen worden lessen daardoor afwisselender, maar ook lijkt het beter om theorie klassikaal in te leiden; leerlingen lezen niet goed als ze achter een computer zitten; de neiging is groot om onmiddellijk commando's te gaan intypen (het verdient daarom ook aanbeveling om, indien mogelijk, niet steeds een computer-lokaal te gebruiken).

Ook is een actieve rol van docenten wenselijk bij het geven van terugkoppeling op het werk van de leerlingen.

De experimentele lessen hadden een nogal vrijblijvend karakter: geen huiswerk, geen overhoringen, geen rapportcijfer. Desondanks werd door de meeste leerlingen goed gewerkt. Een meer officiële status van de lessen zal echter een positieve uitwerking hebben: sommige leerlingen nemen lessen waarvoor ze geen cijfer krijgen onvoldoende serieus, anderen vragen om continuïteit in de vorm van huiswerk, zoals blijkt uit een opmerking die in de evaluatie herhaaldelijk terugkeerde: 'Je kon de stof niet thuis doornemen, en na een week was je alles vergeten.'

Op dit moment vinden vervollexperimenten plaats in 4 havo en 4 vwo. Hoewel de resultaten van deze vervollexperimenten nog niet allemaal beschikbaar zijn, lijken de hiervoor beschreven ervaringen dit jaar bevestigd te worden.

In enkele klassen wordt nu geëxperimenteerd met het ontwerpen van databases. Deze lessen worden gegeven aan leerlingen die eerst de lessen in het bevragen

van een database hebben doorlopen.

Afhankelijk van de uitkomst hiervan zal lesmateriaal ontwikkeld kunnen worden waarin het bevragen en ontwerpen van databases elkaar afwisselen.

Hoe de afwisseling zou moeten zijn tussen ontwerpen en bevragen van databases, zijn op basis van dit lopende experiment nog niet mogelijk.

Noten

- [1] In de kolom KLAS uit de tabel LERAAR staat de klas genoemd waarvan de leraar klasseleraar is.
- [2] Er zijn twee versies van het lesmateriaal gemaakt, die verschillen in de manier waarop de problemen worden opgelost. In de ene versie wordt een probleem opgelost door het globale probleem op te splitsen in deelproblemen die ieder afzonderlijk worden opgelost door verdergaande opsplitsing. De opsplitsing gaat verder totdat tenslotte deel-

problemen zijn ontstaan waarvan de vertaling in computercode aan de ontwerper van de oplossing bekend is. Het hier gepresenteerde voorbeeld is afkomstig uit deze versie van het lesmateriaal.

In de andere versie wordt meteen gestart met het zoeken van delen van de oplossing op detailniveau. Het samenvoegen van deze deeloplossingen tot de eindoplossing vormt de laatste stap in het ontwerpproces.