**DAILY ASSESSMENT FORMAT**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date:** | **10-July-2020** | **Name:** | **Raziya Banu** |
| **Course:** | **Coursera** | **USN:** | **4AL16EC058** |
| **Topic:** | **The Evolution of Data Models** | **Semester & Section:** | **8th sem & ‘B’ section** |
| **Github Repository:** |  |  |  |

|  |
| --- |
| **FORENOON SESSION DETAILS** |
| **Image of session** |
| **Report –** In my first session today I have studied about -The Evolution of Data Models Motivation Relational data models with SQL as data definition and data manipulation language are a quasi-standard for commercially available data base systems, but research in data models and interfaces to database systems seems to diverge into different directions rather than converging to a new, higher-level object model which could serve as a new platform for the non-traditional applications as well as for the classical ones. Object-oriented databases often start from object-oriented programming languages and add persistency as well as other database functions such as transaction management. Objects (instances of abstract data types) are encapsulated and only "accessible" through a well-defined interface. Object manipulation is by invoking type-specific interface functions (methods). The advantages are twofold: first, the structure of objects is hidden (which provides a higher level of abstraction) and second, the methods can implement integrity checks that are specific to the object type.  Typically, OODBMS have a complete programming language environment, but provide little means of descriptive set-operations [GM88]. This reflects their origin from the programming language realm. As there are many programming languages, we expect as many database systems and the integration aspect of databases across several applications written in different languages seems to be lost or at least more difficult. Complex Objects have evolved from relations in that they are constructed by repeated application of tuple and set constructors. Nested Relations as a special case of Complex Objects have been studied 136 in detail in their theoretical as well as practical issues during the past few years [AFS89]. One of the major strong points is the fact that they preserve the descriptive set-oriented style of relational query languages.  Like in many other models, a variety of query languages have been proposed for nested relations and complex objects, such that it might not seem justified to consider this direction as one approach. However, the essential ingredients in all of them are actually the same. Therefore, like in the classical relational model, we can assume one - may be hypothetical - algebra as their common basis. Furthermore, it seems that shared subobjects or recursively structured objects are incompatible with the hierarchical organization and hence can not be brought under the same unifying umbrella. Adding generalization or specialization known from semantic data models [HK87] or similar notions from knowledge representation schemes ultimately seems to increase the divergence.  While such mechanisms facilitate the definition of new types and gives rise to inheritance of attributes, methods, or functions, making query formulation easier, they also need new constructs in queries, such as type predicates and update operations for changing the type of an object. Again, many proposals exist that look pretty incompatible. Furthermore, a number of problems with the formal foundations still lack solutions, for instance how to handle polymorphic types, and how to deal with the higher-order syntactic constructs in a first-order logical framework [Bee89, KL89]. Therefore, in view of all this it appears hopeless to obtain a higher-level unifying data model as a new platform which contains the existing data models as special (degenerate) cases.  However, in this paper we will put together some evidence that the seemingly different concepts are not really incompatible by their very nature. While there exist some differences with respect to formal definitions and in the terminology, we try to show that the essential features are very similar with regard to their behaviour for a user. We think that the differences are not exactly" ... just silly exercises in surface syntax" [SRH90], but we do argue that, apart from the diverging development, we can - in retrospect - interpret them in a consistent way. In order to explain this we describe two evolutionary ways: the first starts with the relational data model and encompasses nested relations, complex objects and object networks ("structural" object-orientation [Dit86]).  The second one starts with the DBTG network model and leads directly to the structural part of object networks. As a result of this exercise we see how recent object models can be obtained as a synthesis of wellestablished concepts, namely  (1) set-oriented, descriptive query and update languages from relations and nested relations,  (2) recursive schema definitions from the network model to allow for object sharing,  (3) behavioral abstraction from abstract data types providing encapsulation (methods), and  (4) inheritance through the specialization of types. Notice that object methods also provide extensibility and adaptation to application classes.  The practical aspect of such a synthesis is that classical data models are contained as special cases. For example standard SQL can be extended to SQL for complex objects. This, in turn, can be generalized to SQL for object networks, and opened to object-specific methods. Thus, a kind of upward-compatibility is achievable which facilitates cooperation between different systems. We do not describe a specific object model here, although we use our notation of [SS90b] and [SS90a] where appropriate, for explanation purposes and as an example.  Our principal goal, however, is to point out how the salient features of recent research proposals are obtained as an evolution from well-known classical ones, and what features have been added to them. In fact, it turns out that apart from all terminological and syntactical differences, a substantial body of core concepts can be found in all of them. The research proposals we have in mind include the ORION query language described in [BKK88, Kim89], the OSQL language of the IRIS project [Bee88, WLH90], the PROBE language [MD86], the 137 object-oriented extension OODAPLEX of the Functional Data Model [Day891, the EXCESS language and EXTRA model of EXODUS [CDV881, the data model used in the O2 project [D+90, LR891, the LauRel project [Lar88], the MAD model [Mit87] with its MQL query language, and the HDBL language [PT86] of the AIM prototype system. Previous work on evolutionary aspects includes [Bee88] and [SS90b].  Theoretical work on the foundations of OODB models [AK89, Bee89] shows strong similarities to nested relational/Complex Object models. The same is true for query languages/algebras developed for 00 models [Kim89, SZ89]. In the following Section 2 we describe the evolution starting with relations and Section 3 contains the evolution when the DBTG network model is used as a starting point. In Section 4 we show that similar results could be obtained when starting from models such as the Functional Data Model, various semantic data models, or knowledge representation languages. This section also contains some remarks on the inclusion of (retrieval and update) methods, that is, "behavioral" object-orientation. Finally we conclude in Section 5 with a summary. 2 Extending Relations to Objects 2.1 From Relations to Nested Relations and Complex Objects When we consider the relational model as a language for defining data types, that is, given some set of base types (the atomic domains) we construct relations from these, we find that the only type constructor available is relation. Relations are sets of tuples, however, in contrast to the set (@) and tuple (0) constructors found in semantic data models [HK87], the relational model does not offer them as separate constructors.  Further, the first normal form condition requires that the domains of attributes (that is, the tuple-components) be atomic. Therefore, this relation type constructor can only be applied once per constructed type (relation). The nested relational model [SS86] is obtained from the flat relational one by allowing relations as values of attributes. That is, we allow the single type constructor relation to be applied any number of times in a constructed type (nested relation). If we further relaxed the restriction that set and tuple constructors have to strictly alternate, then we end up with what is usually called "Complex Objects" [AB88) or extended nested relations in [PT86]. Figure 1 gives an overview of (INF-) relations, nested |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date:** | **10-July-2020** | **Name:** | **Raziya Banu** | |
| **Course:** | **Cisco** | **USN:** | **4AL16EC058** | |
| **Topic:** | [**Educational and Business Opportunities**](https://static-course-assets.s3.amazonaws.com/I2IoT20/en/index.html#6) | **Semester & Section:** | **8th sem & ‘B’ section** | |
| **AFTERNOON SESSION DETAILS** | | | |
| **Image of session** | | | |
| **Become an Informed Consumer**  The last few years have given us improvements in the speed and availability of Internet services, as well as advances in cloud computing and sensor technology. These technical gains, together with recent developments in automation and artificial intelligence, have created a highly digitized world. Digitization currently impacts every aspect of our daily lives. Digitization continues to provide new opportunities for professionals who are trained to develop and support the technology that is used to deliver the IoT.  The IoT provides an immeasurable amount of information that is readily available for consumption. This information can be quickly analyzed and used to automate many processes that were previously considered impossible to turn over to machines. For example, just a few years ago self-driving cars existed only in our imaginations and now they are a reality. Think about what else has changed in your life because of the IoT.  The IoT is also freeing humans from the drudgery of routine and repetitive tasks such as restocking shelves and order fulfillment. We may now have more time for higher intellectual pursuits and the chance to explore all the IoT has to offer. **Challenges in the Digitized World** The IoT provides many benefits but at the same time it presents many challenges. Since the IoT is a transformational technology, we are now faced with an ever expanding collection of new technology that we must master. The IoT is changing every aspect of our lives.  This is not the first time we have experienced a technological development that has such an impact. Mechanization on the farm allowed increased productivity of available farmland and started the migration of the population from rural to urban areas. The development of the automobile allowed for greater mobility of the workforce and increased recreational activities. The personal computer allowed the automation of many routine tasks with improved accuracy and efficiency. The Internet started to break down geographic barriers and improve equality between people on a global scale. These are only a few of the transformational technologies that we have experienced in recent history.  Every one of these technologies presented major changes to an established society and was met with initial fear and apprehension. After the initial fear of the unknown was overcome and the technology was embraced, the inherent benefits became obvious. Each perceived challenge opens up many new opportunities. **Entrepreneurs needed!** The IoT is also creating a demand for a new kind of IT specialist. These are individuals with the knowledge and skillsets to develop new IoT-enabled products and process the data they collect.  An entrepreneurial workforce is needed that specializes in both information science and software or computer engineering.  Additionally, operational technologies and information technologies are converging in the IoT. With this convergence, people must collaborate and learn from each other to understand the things, the networks, and methodologies that harness the limitless potential of the IoT. **Lifelong Learning** With the ever changing landscape of the digitized world, we must stay current in order to realize the full potential of what the IoT has to offer.  The job market will continue to offer more opportunities as new technologies evolve. The skill sets required for these jobs will evolve at the same time, thus creating the need for lifelong learning. **Cisco Networking Academy** The rapid growth of networks has created a global shortage of people who are qualified to implement and maintain networking solutions, especially in places where networks are being built to promote economic development. At the same time, people need access to better training and career opportunities to successfully compete in the global economy.  With over 10,400 academies in 180 countries, the Cisco Networking Academy helps individuals prepare for industry-recognized certifications and entry-level information and communication technology (ICT) careers in virtually every type of industry. The Cisco Networking Academy helps address the growing demand for ICT professionals, while improving career prospects in communities around the world.  The Cisco Networking Academy Program has trained more than five million students to date. Many graduates have gone on to successful IT careers in a variety of industries, while others have harnessed the entrepreneurial spirit and knowledge they acquired to start their own businesses and create new jobs. **Networking Academy Curriculum** The Networking Academy delivers a comprehensive, 21st century learning experience. Students develop the foundational IT skills needed to design, build, and manage networks, along with career skills such as problem solving, collaboration, and critical thinking. Students complete hands-on learning activities and network simulations to develop practical skills that will help them find their place among networking professionals around the world. These are some of the offerings of the Networking Academy:   * **IoT Fundamentals** – This series of courses teaches you about the IoT and how it can be used to enhance society. This series continues to evolve. It currently includes courses and activities to develop your skills for securely collecting data and connecting sensors to the cloud, analyzing big data, and creating your own IoT solution. * **IT Essentials**- IT Essentials covers the fundamentals of computer hardware and software. It also introduces more advanced concepts, such as security, networking, and the responsibilities of an IT professional. * **Entrepreneurship**- The Entrepreneurship course teaches critical business skills, financial skills, attitudes, and behaviors to help students develop an entrepreneurial mindset which can empower them to improve their overall quality of life. * **Introduction to Cybersecurity** - The Introduction to Cybersecurity course covers trends in cybersecurity and demonstrates the need for cybersecurity skills in various industries. * **CCNA Routing and Switching** – Cisco Certified Networking Associate (CCNA) Routing and Switching provides a comprehensive overview of networking concepts and skills. It covers skills and knowledge required for administrators of devices in small to medium-sized networks. This curriculum has an emphasis on practical application, work-force readiness, and soft-skills development. * **CCNA Security** - CCNA Security introduces the core security concepts and skills needed to install, troubleshoot, and monitor a network to maintain the integrity, confidentiality, and availability of data and devices. * **CCNP –** The Cisco Certified Networking Professional (CCNP) curriculum is the next step for people who have completed the CCNA Routing and Switching courses.  **Industry Certifications** Industry certifications are highly respected by employers around the world and help validate the skills needed to launch successful careers in networking and IT. Certifications are achieved by passing an exam proctored by a certifying authority. Students must complete training materials specific to the certification exam. Field experience is often very helpful, but not always required, to pass a certification exam. Cisco Networking Academy provides courses that prepare students for the industry certifications that are shown in Figure 1.  There are two basic types of certification available: vendor-specific and vendor-neutral. Vendor-specific certifications are tailored to technologies offered by a company to prove that an individual is qualified to deploy and manage that technology. Vendor-neutral certifications are offered by many different organizations. They show that an individual has a well-rounded skillset centered on common systems and programs, rather than specific brands of technology.  Most often, certifications must be renewed over time. Requirements for re-certification may be earning continuing education units (CEUs), passing a re-certification exam, or both. CEUs can be earned by attending classes, professional membership, on-the-job experience, or research and publishing of materials that support the certification technology. **Additional Learning Resources** Certifications can show an employer that an individual has the appropriate skills for a job. Community college or university degrees can show that a person has gained a broad understanding in a field of study. This broad understanding creates a solid foundation for emerging career opportunities in the IoT. A combination of industry certifications and university degrees provides a student with the best background, experience, and education to pursue a career with greater opportunities and higher salary.  When looking for a degree to pursue at a community college or university that will pertain to the skillsets needed for a career in the IoT, watch for some of the following degrees:   * Business Intelligence * Computer Information Systems * Computer Programming * Computer Science * Database Administration * Electromechanical Automation * Electronics Engineering * Linux Networking * Machining * Network Administration * System Analysis * Web Server Administrator   This is not an exhaustive list. Even traditional degree programs such as supply chain management, business, and project management are helpful for careers in IoT. Computer-Aided Design (CAD), drafting, math, and physics are applicable and show a diverse education, which is perfect for an IoT career. | | | |